FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Miniature Robotic Manipulator for Remote Chemistry Laboratory

Érica da Cunha Gomes

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Armando Jorge Miranda de Sousa

Second Supervisor: Paulo José Cerqueira Gomes da Costa

July 19, 2018

Resumo

Experiências laboratoriais de química nas aulas do ensino secundário são dispendiosas, requerem muito tempo e o seu acesso é limitado ao horário de aulas.

É interessante tirar partido das vantagens oferecidas pela robótica e pelo controlo digital remoto para permitir que os estudantes tenham um maior acesso às experiências, podendo assim, ter uma maior facilidade de aprendizagem. Além disso, é também possível ultrapassar dificuldades que os professores possam encontrar na hora de tentar realizar as experiências nas suas aulas. Foram problemas desta natureza que levaram à criação do "FEUP ChemLab", um sistema inovador que permite, por exemplo, estudantes e professores do ensino secundário ter acesso, monitorizar e controlar verdadeiras experiências à distancia usando ferramentas digitais comuns (*web browser*). O "FEUP ChemLab" inclui um manipulador robótico e uma *webcam* que permite aos estudantes finalizar uma experiência que tenha sido iniciada mais cedo na sala de aula.

O *hardware* do sistema conta com seis motores, três *encoders* magnéticos, dois sensores de força, um acelerómetro, quatro microcontroladores, três sensores de fim de curso e foi parcialmente impresso por uma impressora 3D. O *software* foi criado usando as linguagens: *C, PHP, HTML/ CSS, JavaScript, Lazarus/FreePascal*.

É utilizado o *"uStepper Robotic Arm"* com quatro graus de liberdade é usado. Um *end effector* com uma dupla garra e rotação de pulso foi desenhado e implementado. O *end effector* é capaz de agarrar objetos com uma força controlada, de transferir e pipetar líquidos.

A comunicação entre os microcontroladores, o servidor *web* e a base de dados utiliza mensagens UDP, *mySQL queries* e comunicação porta série sob barramento USB.

No *website* são guardadas posições predefinidas com os ângulos de cada motor. Ao clicar nos botões disponíveis correspondentes às posições o utilizador consegue controlar o manipulador robótico e assim realizar uma experiência e ver resultados através da *webcam*.

Abstract

Chemistry laboratory experiences for high schools are expensive, time consuming and access is generally limited to school time.

It is useful to take advantage of the interest of robotics and remote digital operation to give further access to students to enhance learning and surpass the difficulties teachers encounter when trying to implement experiments in their classes. Such issues led to "FEUP ChemLab", an innovative system that allows high-school students and teachers to access, monitor and control a real experience at a distance using common digital tools (web browser). The "FEUP ChemLab" includes a robotic manipulator and a webcam that allows, for example, students to finish a experiment they have started earlier in the class.

The hardware of the system includes six motors, three magnetic encoders, two force sensors, one accelerometer, four microcontrollers, three limit switches and was partially 3D printed. The software was written in C, PHP, HTML/CSS, JavaScript, Lazarus / FreePascal.

The uStepper Robotic Arm with four degrees of freedom is used. A new end effector with dual grippers and wrist rotation was designed. The system is capable of force control gripping, transferring and pipetting liquids.

The communication between the microcontrollers, the web server and the database is implemented using UDP messages, mySQL queries and serial port communication under USB connection.

Predefined positions with the angles for each of the motors are saved in the website. By clicking on available buttons corresponding to positions, the user controls the robotic manipulator and is able to perform an experiment and see the results using the webcam.

Acknowledgments

Antes de mais, quero agradecer aos meus orientadores, Professor Armando Sousa e Professor Paulo Costa, por toda a ajuda, tempo, paciência e orientação durante este ano.

Agradeço também a todas as pessoas que de qualquer maneira ajudaram na realização deste enorme projecto.

À minha mãe e aos meus avôs por todo o apoio e amor durante os meus curtos 22 anos. Sei que não seria a pessoa que sou hoje sem vocês e não teria chegado aqui.

Aos meus amigos por serem os maiores idiotas do planeta, pelas vezes que não fazem sentido, pelas vezes que não estamos de acordo, por todas as vezes que me corrigiram, pelas mil e uma dores de cabeça e por todas as vezes que irão repetir isto no futuro. Nada nestes 5 anos teria tido metade da magia que teve sem vocês.

Finalmente, o maior obrigada à melhor pessoa. Nunca te puderei agradecer o suficiente pela pacicência de cada vez que desanimava e ia abaixo, pelas surpresas, por todos os conselhos, por todas as vezes que me ouviste mesmo quando era a última coisa que precisavas, por toda a ajuda e principalmente por estares aqui todos os dias...

Érica Gomes

"Theory is when you understand everything but nothing works. Practice is when everything works but you don't understand why. In this place we combine theory with practice: nothing works and no one knows why."

Unknown

Contents

1 Introduction **1**
1.1 Context and Motivation 1
1.2 Goals 1
1.3 Scope 2
1.4 Structure of the Document 2

2 Laboratories and Technologies **3**
2.1 Laboratories 3
2.1.1 Hands-on Laboratories 3
2.1.2 Virtual Laboratories 3
2.1.3 Remote Laboratories 5
2.1.4 Conclusions 6
2.2 Technologies 6
2.2.1 Robotic Manipulator 6
2.2.2 Computer 10
2.2.3 Camera 12

3 Requirements and Architecture **15**
3.1 Requirements 15
3.1.1 Robotic Manipulator Requirements 16
3.1.2 Web Interface Requirements 16
3.2 Architecture 17

4 Web Interface **19**
4.1 Website 19
4.1.1 Home Page 21
4.1.2 Profile Page 23
4.1.3 View Mode Page 29
4.1.4 Control Mode Page 30
4.2 Database 37

5 Robotic Manipulator **39**
5.1 Body of the Manipulator 39
5.2 End Effector 45
5.3 Connecting the End Effector to the Body 50

6 Communication Platform **53**

7 Conclusions and Future Work **57**
7.1 Compliance with the requirements 57
7.2 Conclusions 58
7.3 Contributions 58
7.4 Future Work 59

A Experiment's Protocol **61**

References **65**

List of Figures

2.1 AL5D PLTW Robotic Arm . . . 7
2.2 SSC-32U Servo Control Boar . . . 8
2.3 uStepper Robotic Arm . . . 9
2.4 uStepper Board and NEMA 17 stepper motor with the uStepper attached . . . 10
2.5 Raspberry Pi 3 Model B . . . 12
2.6 Hue HD Camera . . . 13

3.1 Final Architecture . . . 17

4.1 Representation of the LAMP model. . . . 19
4.2 Website's map. . . . 20
4.3 Home page. . . . 22
4.4 Log In page. . . . 22
4.5 Password recovery page. . . . 23
4.6 Password recovery email. . . . 23
4.7 Administrator's profile page. . . . 24
4.8 Sign Up page. . . . 24
4.9 List of users page. . . . 25
4.10 Change password page. . . . 26
4.11 Schedule and list of schedules page. . . . 26
4.12 Remove an User page. . . . 27
4.13 Page the user is redirect to after a log out is successful. . . . 28
4.14 Student's profile page. . . . 28
4.15 Group's details page. . . . 29
4.16 View mode page. . . . 30
4.17 Control mode page. . . . 30
4.18 Sequence diagram for the control page. . . . 31
4.19 Logical actions. . . . 34
4.20 Waiting page for a teacher. . . . 36
4.21 Waiting page for a student. . . . 36
4.22 Representation of the database. . . . 38

5.1 Base of the robotic manipulator with labels. . . . 40
5.2 Representation of the possible movements and the responsible motors. . . . 40
5.3 Modeled lateral view of the gears of the motor 0. . . . 41
5.4 Encoder problem representation for the motor and uStepper board 0. . . . 41
5.5 Encoder angle transformation for the motor and uStepper board 0. . . . 42
5.6 Encoder angle transformation for the motor and uStepper board 2. . . . 43
5.7 Encoder angle transformation for the motor and uStepper board 1. . . . 44

5.8 3D printed piece and limit switch for the motor 1. 44
5.9 Original gripper. 45
5.10 New end effector. 45
5.11 De-constructed 3D model of the end effector's pieces. 46
5.12 YFRobot Sensor Shield. 46
5.13 Adafruit Motor Shield V2. 47
5.14 HX711: analog-to-digital converter. 47
5.15 Architeture for the new end effector. 48
5.16 Position controller for the wrist rotation. 49
5.17 Algorithms for the three different working modes of the grippers. 49
5.18 View of the springs on the robotic manipulator from the back. 50
5.19 View of the springs on the robotic manipulator from the side. 50

6.1 Representation of the communication sequence. 54
6.2 Interface of the communication platform for the uStepper 0. 55
6.3 Interface of the communication platform for the end effector. 55
6.4 Representation of the connections between each element. 56

List of Tables

2.1 Examples of remote laboratories using robotic manipulators 6
2.2 AL5D PLTW Robotic Arm Dimensions 7
2.3 SSC-32U Servo Control Board Specifications 8
2.4 uStepper Robotic Arm Dimensions 9
2.5 Some uStepper and NEMA 17 Specifications 10
2.6 Intel NUC Board DE3815TYBE Specifications 11
2.7 Raspberry Pi 3 Model B Specifications 11

4.1 Meaning of each letter and number of the position code. 32
4.2 The three sets of the form options to make a sentence. 33
4.3 Physical interpretation (with the meaning of the code in the table 4.1). 34

7.1 Approximated cost and weight of each component. 58

Abreviaturas e Símbolos

ADC	Analog-to-Digital Converter
CSS	Cascade Styling Sheets
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
JS	JavaScript
LAMP	Linux, Apache, MySQL, PHP
LAN	Local Area Network
NUC	Next Unit of Computing
PHP	Hypertext Preprocessor
PWM	Pulse Width Modulated
RAM	Random Access Memory
SPI	Serial Peripheral Interface
STEM	Science, Technology, Engineering and Mathematics
UDP	User Datagram Protocol
USB	Universal Serial Bus
UVC	USB video class
VGA	Video Graphics Array

Chapter 1

Introduction

1.1 Context and Motivation

Schools and the Government are seeking to introduce robotics to students of several ages[1]. Robotics are not only learning objects but also learning tools. As a learning tool, robots help get students more interested in Science, Technology, Engineering, and Mathematics (STEM) subjects. In a study, students stated that, when interacting with robots, they are not only learning but also feel like they are playing, making them want to participate and study more. There are several studies proving this and more positive impacts in the classrooms [2].

Chemistry teachers suffer many setbacks when trying to implement experiments in their classes. The large number of students per classrooms, the insufficient material available in the schools' laboratories for so many students, undisciplined students that prevent from doing safe experiments and the lack of time to finish the laboratory activities are some of the reasons [3].

Therefore, there is a need for a solution that overcomes this difficulties. This need and the newfound enthusiasm for robotics in classrooms has led to the creation of this project.

1.2 Goals

The main objective of this project is to create a innovating system that allows high-school students to access a remote laboratory in order to complement the learning process, for example, to finish an experiment students have started earlier in the class. The aim is to create a system that any student with a device connected to the Internet can use. It needs to be simple enough for easy understanding but also needs to be safe in other to minimize accidents with the experiments. It's also important for the system to be low cost so schools can afford to obtain it and introduce it to daily classes.

A secondary goal is to validate the system by performing a Green Chemistry experiment. With the increased awareness of the global warming and the wish to decrease pollution, Green Chemistry has a higher role in the industry and the education of the future generation. There is hope for a better educated population regarding the way the planet is treated. Green Chemistry can be defined has "the use of chemistry for pollution prevention. More specifically, it is the design of chemical products and processes that are environmentally benign" [4]. However, it is not only applicable in the industry but also in everyones daily tasks. With this in mind, it's important to create opportunities for students to increase engagement and the will to learn more about this so they become more informed and able to make better decisions.

Finally, with this system it's also intended to create awareness in high school students for robotics and robotic applications increasing student's engagement in STEM subjects.

1.3 Scope

In this project, the hardware components to be used, such as the body of the robotic manipulator and the camera, already exist in the market, removing the need to create new ones from scratch.

Being this said, the focus is centered on controlling the robotic manipulator, the connections and communication between each component, creating a friendly user interface and making small necessary changes to hardware.

1.4 Structure of the Document

This document is divided into seven chapters, including the present chapter. In chapter 2 a review of different types of laboratories and different technologies considered for the implementation of this project are presented. Next, in chapter 3, the architecture and an overview of the system are introduced as well as requirements for both the manipulator and the interface.In chapter 4 the implementation of all the parts of the interface and how it works is elaborated. In chapter 5 alterations made to the original robotic manipulator and it's workings are explained. After this, in chapter 6 it's described how the communication between the robotic manipulator and the web interface is made. Finally, in chapter 7 conclusions are taken and future work is established.

Chapter 2

Laboratories and Technologies

2.1 Laboratories

2.1.1 Hands-on Laboratories

A hands-on laboratory is a physical real laboratory where investigation and experimental processes happen. What distinguishes this type of laboratory is having all the equipment required and the person using the the equipment physically in the same space [5]. However, these laboratories can be very expensive due to the need of an expert when students are the ones using it, maintenance and space.

2.1.1.1 Green Chemistry Experiment in Hands-on Laboratory

To test the functionality of the system, a green chemistry experiment was chosen: Synthesis of Iron(II) Oxalate dehydrate. This experiment was selected in order to prove that the system developed is a functioning solution meeting all the requirements.

In the appendix A the protocol for this experiment, developed by the *Centro de Química da Universidade do Porto*, can be found [6]. This protocol is meant for a hands-on laboratory.

To execute this experiment with the system some little changes were done. The steps 1 and 2 will be completed by the students and the teacher previously in the class. Then, the next steps will be performed by the robotic manipulator controlled by the students remotely. In order to simplify the process, the step 4 will only correspond to the agitation of the mixture for 10 minutes.

2.1.2 Virtual Laboratories

A virtual laboratory or a simulator is a fully software based application capable of performing a virtual experiment. In a virtual laboratory is important for the simulated tools and the space to feels as authentic as possible [7].

While programming this kind of system, it's necessary to keep in mind what is being simulated in order to make sure the system has the same behavior, reactions and results as the real one. For this, exhaustive research on the system to be simulated needs to be done, from talking to the manufactures to examining all the small details [7]. The time it takes to run a simulation is also very important. It either takes the same time as in reality, giving more emphasis to the process time; or it's faster than in the real world, allowing the user to focus more on the results.

Many reasons can be found for the appearance and use of virtual laboratories. Economic reasons seem to be very strong, as by using simulators the need for a big physical laboratory with capability for many users decreases [8] [5]. But the advantages don't stop there [7]:

- **Flexibility** Different experiments with different components and variables can be easily performed in the same application.
- **Multiple access** Different users can execute an experiment using the same simulation at the same time.
- **Configuration Parameters** The settings are easily and quickly changed in the virtual laboratory unlike in a physical laboratory.
- **Damage Resistance** Users can repeat experiments without fear of damaging the materials.
- **Repetition** Infinite repetition is possible, allowing for the user to observe multiple experiments no matter the final result of the previous experiment performed.
- **Unobstructed View** A clear line of sight of all angles is available as there are no devices protecting the experiment. In a physical laboratory the view can be reduced due to protection such as a dust cover or a fume extractor.

However, some drawbacks are associated with simulations.

- **Cost** One concern, contradicting what was previously said, is that the actual cost of a realistic simulation isn't lower than a real laboratory; which is explained by the fact that a good virtual laboratory needs a lot of time and consequently money to be accurate [5].
- **Lack of consequences** Another problem comes from the fact that the system actually doesn't exist and therefore can create informality in a sense that the users show irresponsibility and negligence that otherwise, in a real laboratory, they wouldn't have. Which can as well be explained by never experiencing the consequences of a wrongly performed experiment in reality [7].
- **Resources** The need for specific resources and high performing computer in some cases can be an obstacle [7].
- **Incomplete skills** There is a big distance from the reality that is present in a simulator and, in the end, a hands-on laboratory is always necessary to teach some basic and crucial skills [7] [5].

2.1.3 Remote Laboratories

A remote laboratory enables distant access to a real and physical laboratory [7]. In a remote experiment the user is controlling physical means using a computer as an intermediary. This allows for the user to be anywhere as with a simulation but still be able to receive real values and obtain real data as in a hands-on laboratory [8].

In a simple way, a remote laboratory is based on an interaction between the client and the server. In one end, a computer allows for the control of the experiment, accordingly to its needs, and connectivity to the Internet serving as a server. In the other end, the user will access an interface (such as a web browser) in order to give control commands and supervise the experiment [9]. It's important that this interface is easy to access, available for all the users and most used devices.

The advantages associated with remote laboratory are numerous:

- **Collaboration** It's possible to share the physical laboratory with other entities and consequently the data and results of different experiments and points of view expanding knowledge [5].
- **Increased usage** Because it's accessing a hands-on laboratory remotely, it increases the number of times a user can perform an experiment as it's not constrained by the laboratory open hours and available 24 hours, 7 days a week [5].
- **Return of investment** As it can be used more often and by different institutions, the investment return on the laboratory is higher [9].
- **Improved inclusion** For disabled users who can't access the laboratory or operate the equipment in a traditional way, this laboratory offers a solution [10].
- **Prevent overcrowding** When the laboratory is small or when there is a high number of users at the same time in the same space, using the remote laboratory offers a more efficient use of the space [10].

On the other hand, the following disadvantages are associated with this laboratories:

- **Slow functionality** Users can get frustrated and impatient as it can take a little bit longer in some actions or distracted with the computer [5].
- **Cost** A Remote laboratory is very expensive as it needs the same physical features as a hands-on laboratory plus an actuator and the software (usually simpler than a simulator) on the other end.
- **Other similarities to Hands-on Laboratories** It shares some other disadvantages with the Hands-on Laboratories such as the need for a technician.

Table 2.1: Examples of remote laboratories using robotic manipulators

Name	Function	Manipulator	Year
The Mercury Project [11]	Study - First remote manipulator	- Made -	1994
Ciclope Robot [12]	Education - University	Fisher-Technik	2009
ILab [13]	Education - University	RA-01 Robotic Arm	2009
Lynxmotion AL5 in [14]	Personal use	Lynxmotion AL5	2013
Robot Arm in [15]	Industry	Erik - Robotnik	2014
Remote Laboratory in [16]	Education - University	NND - Nakanippon Electric Ldt.	2015
Robotic Articulator in [17]	Education - University	- Made -	2016

2.1.4 Conclusions

Each laboratory type presented has its own advantages and disadvantages, however what's important is to notice that all the above have a similar success when it comes to its learning outcome. [8] It's also possible to create a hybrid laboratory joining a virtual and a remote laboratory in one. This combines the advantages of the laboratories with an expected increase in the cost. When choosing the type of laboratory it's important to focus on what is needed and what is the end goal. This is why a remote laboratory was chosen to implement in this project.

In the table 2.1, some examples of remote laboratories using robotic manipulators are presented and a lack of remote laboratories for high school education can be found. This is one of the reasons this project is so important.

2.2 Technologies

In this section, some technologies needed based on an initial concept are presented and compared.

A robotic manipulator is controlled trough a microcontroller which is connected to a computer. The microcontroller is responsible for reading and setting the inputs and outputs while the computer is responsible for processing intensive tasks. The computer is also responsible for hosting the web server and the data base. A camera is connected to the computer so the video stream can be sent to the web application.

In the next sections, the options considered in order to choose the best components to use in the system are presented.

2.2.1 Robotic Manipulator

Two robotic manipulators were initially considered: the AL5D PLTW Robotic Arm and the uStepper Robotic Arm. The final choice was the uStepper Robotic Arm because the control of the motors is more suitable for the task ahead as is discussed in the next points.

2.2.1.1 AL5D PLTW Robotic Arm

The AL5D PLTW Robotic Arm, shown in the figure 2.1, is a 4 degrees of freedom robotic manipulator that costs USD358. All the information about this manipulator can be found on the manufacturer website in [18]. This manipulator was developed for the Project Lead The Way (PLTW), a STEM solution that is used in over 5000 schools in the United States of America. A software was created so students can have full control of the arm without having to spend a lot of time programming.

Figure 2.1: AL5D PLTW Robotic Arm. Source [18]

The structure of the robotic arm is composed of aluminum brackets, aluminum tubing and hubs, custom injection molded components and laser-cut components. However, the arm is “hobby grade” and should not be used to lift heavy objects. The maximum tested load is less than 300 mL at full reach without any additional products. The lift capacity increases as the load is closer to the base, but it depends on the configuration of the arm. [19]

Table 2.2: AL5D PLTW Robotic Arm Dimensions

Distance - base to elbow axis	14.6 cm
Distance - elbow to wrist axis	18.7 cm
Height - arm parked	18.4 cm
Height - reaching up	48.3 cm
Median forward reach	51.5 cm
Gripper opening	3.2 cm

This robotic manipulator uses Hitec RC servo motor actuators at each joint. This motors have gears that increase torque and are blocked mechanically to rotate only 180°, making small movements. They also have a small control circuitry which has three signals: 5V, ground and Pulse

Width Modulation (PWM). Through the PWM, a signal from the servo control board SSC-32U is received allowing the motor to move at the same duty cycle as the PWM signal that was set. It's important to notice that the servo motors are not meant to operate for extended periods of time as they will overheat. The duty cycle is 25%. This means that in a period of one hour, for example, the arm should only be operating for around 15 minutes continuously. The rest of the time the arm should not be functioning and the motors are left to cool down.

The SSC-32U board, shown in the figure 2.2 is a dedicated servo controller board that allows controlling up to 32 servos channels with 1µS resolution and access to 8 analog input/output pins that allow for a query of the sensor values. The motion control can be immediate response, speed controlled, timed motion, or a combination. A function "group move" allows for a combination of any servos to begin and end motion at the same time, even if the servos have to move different distances. The servos' position or movement can be queried to provide feedback to the host computer.

Table 2.3: SSC-32U Servo Control Board Specifications

Microcontroller	Atmel ATMEGA328P
PC interface	USB mini B
Servo travel range	0.5ms to 2.5ms
Servo resolution	1µS (0.09° for 180° servos)
Servo motion control	Local closed loop
VS peak current	max 15A per side
VS steady current	max 3-5A per side

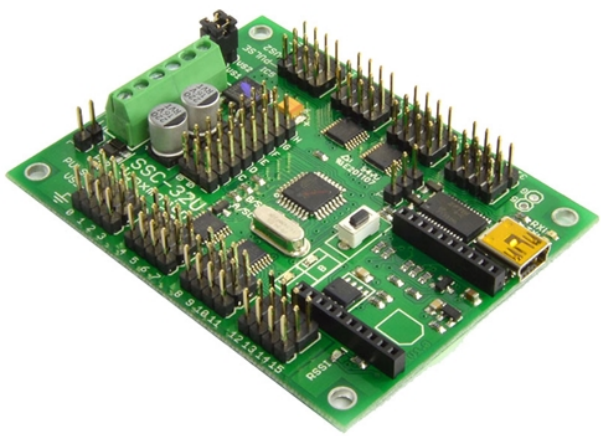


Figure 2.2: SSC-32U Servo Control Board. Source [20]

2.2.1.2 uStepper Robotic Arm

The uStepper Robotic Arm, shown in the figure 2.3, is a three axes open source robotic manipulator developed by ON Development in Denmark. All the informations about the manipulator, its parts and functioning can be found on the manufacturer's website in [21] and [22]. This manipulator is suitable for learning about robotics and programming as it's fairly simple, already comes with code developed and costs around 364 euros (without taxes) for the complete kit.[21]

Figure 2.3: uStepper Robotic Arm. Source [23]

The parts that constitute the robotic arm are 3D printed with a resolution of 0.2 mm and 25% infill and some aluminum tubes. The full Stereolithography (STL) design files are available at the uStepper GitHub repository [22] for free community use. This makes it easier to make spare parts or print the full manipulator to possibly reduce costs. Although, the quality of the parts is considered to be good with the print settings referred above, it should be noticed that it is not comparable to injection mold parts. The maximum lift possible is of 350g when furthest to the base and 700g when close to the base.

Table 2.4: uStepper Robotic Arm Dimensions

Distance - base to elbow axis	32 cm
Distance - elbow to gripper axis	22 cm
Height - arm parked	30 cm
Maximum Rotation Angle	360°

The uStepper Robotic Arm uses NEMA 17 stepper motors instead of an DC motor, which provides a high holding torque without fluctuations in the shaft position. Stepper motors are precise which is why they are often used when it is necessary to move something accurately to a given distance. Also, they are very cheap when compared to servo motors.

The uStepper Robotic Arm uses uSteppers boards. This is a microcontroller with inputs and outputs, a stepper driver and an encoder all together in a compact board that fits on the back of the NEMA 17 steppers, as shown in the figure 2.4. The compactness of the board enables the development of applications without the need for long and messy wiring to an external Arduino. The uStepper is capable of solving the problem of not knowing if the steppers motors are moving. This is, if a stepper encounters for some reason an obstacle it will not take it into account and will not correct it. The uStepper solves it by continuously monitoring where the motor is, where it should be and compensating if needed. The position tracking is possible due to the 12-bit rotary encoder, meaning that the shaft position can be tracked in steps of 1/4096, corresponding to a resolution of 0.088°.

Finally, to make it easy to program, the uStepper is compatible with the Arduino Integrated Development Environment (IDE).

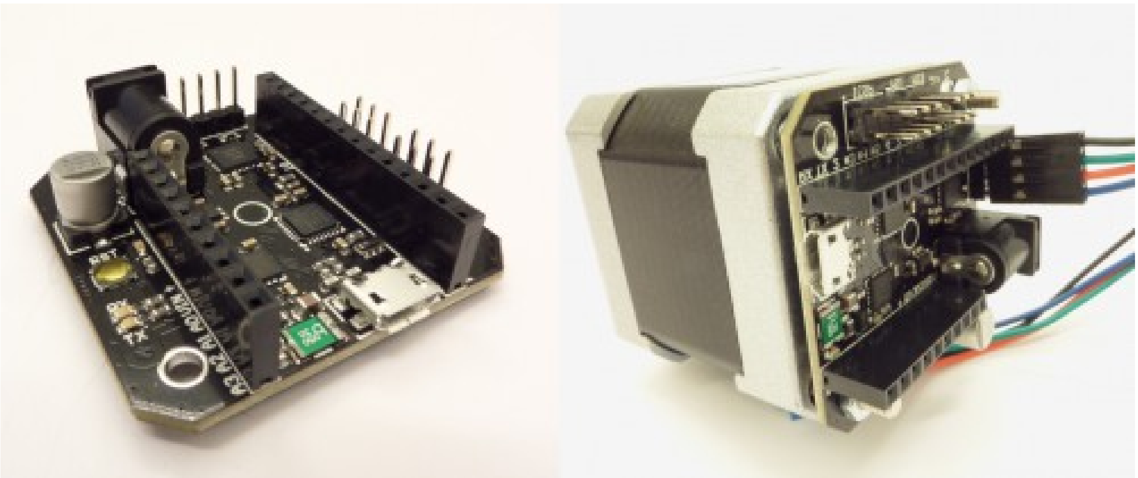


Figure 2.4: uStepper Board and NEMA 17 stepper motor with the uStepper attached. Source [21]

Table 2.5: Some uStepper and NEMA 17 Specifications

Microcontroller	Atmel ATMEGA328P
PC interface	USB
Digital I/O pins	12
Analog I/O pins	4
Holding Torque	4200 g.cm
Step angle	1.8°
Encoder resolution	12-bit
Microstepping	up to 16x

2.2.2 Computer

An affordable and efficient computer is necessary to process tasks. In the next points, two options are discussed.

2.2.2.1 NUC

A NUC (Next Unit of Computing) is a small computer (around 10cm by 10cm) produced by Intel. There are three types of NUC and several versions of each: mini PCs, boards or kits. The mini PCs are ready to work computers with Windows 10 operating system and range from 211$ to 899$. The kits are customizable boards prepared for different memory, storage and operating systems ranging from 122$ to 689$. The boards come with a processor but allow for a total customization ranging from 119$ to 398$. [24] The NUCs are a good option because they have around the same power as middle tear laptop, depending on the version, but are much smaller, lighter and customizable.[25] However, the high price compared to the next option and the lack of general purpose input/output (GPIO) pins in most models are some negative aspects. Some relevant specifications from the cheapest NUC, the Intel NUC Board DE3815TYBE, are presented in the table 2.6.

Table 2.6: Intel NUC Board DE3815TYBE Specifications

Processor	Intel Atom Processor E3815
RAM	Maximum 8GB
Ports	HDMI, VGA, 5 x USB 2.0, 1 x USB 3.0, Ethernet
Integrated Wifi	No
Integrated Bluetooth	No

2.2.2.2 Raspberry Pi

The Raspberry Pi, pictured in the figure 2.5, is a credit card-sized open hardware computer slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level. One of the main advantages of using a Raspberry Pi is the size of the community and the amount of open and documented projects available. [26] Another benefit is the low price as the latest model, the Raspberry Pi 3 Model B, costs 35$. This model has wireless local area network (LAN) and a high number of GPIO pins which are desired features. The Raspberry Pi uses a Raspbian Linux distribution as the operating system.[27] In table 2.7 some relevant specifications of the latest model of the Raspberry Pi are presented.

Table 2.7: Raspberry Pi 3 Model B Specifications

Processor	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
RAM	1GB
Memory	Micro SD card
GPIO	40-pin
Other Ports	HDMI, 4 x USB 2.0, Ethernet
Integrated Wifi	Yes
Integrated Bluetooth	Yes
Power source	Micro USB 2.5A

Figure 2.5: Raspberry Pi 3 Model B [27]

2.2.3 Camera

A camera is required in order to video stream the bench work to the web application so the user can see what he is doing. Since there are thousands of cameras in the market to chosen from and many of them similar, there is no relevancy in comparing.

One of the requirements demanded is that the camera supports USB video class (UVC) standard. UVC is a standard way to represent video from a web camera or a digital video camera to another device such as a display or a computer through USB. UVC standardizes the video drivers and is built into Windows, Mac OX, Linux, FreeBSD an the Playstation 3. This way, when a web camera is plugged using USB into a computer (for instance) it is immediately recognized as a video source. [28]

The selected camera is the Hue HD camera, shown in the figure 2.6 because it meets the UVC requirement and has several appealing features. The camera has an adjustable flexible neck and can stand on base that connects to a computer or be plugged in directly into an USB port. It's UVC compatible with Windows (XP, Vista, 7, 8, 10) Mac OS X 10.4.3 and later version and with most Linux distributions. The camera has a viewing angle of 73°, a focus range from 5 cm to infinity and an integrated microphone[29].

Figure 2.6: Hue HD Camera

Chapter 3

Requirements and Architecture

3.1 Requirements

The requirements exist to determine what the system needs to do and how it should be done. They are usually determined along with the client or an expert in the area of work.

In order to get the requirements for this project, a meeting with Professor Carla Morais, an expert in educational chemistry and teacher in the Faculty of Sciences of University of Porto, and Professor Armando Sousa, an expert in robotics and teacher in the Faculty of Engineering of University of Porto, was held. From this meeting it was decided that the best solution for this project is a remote laboratory with specific characteristics.

Firstly, students should be able to control a robotic manipulator through a user interface (a website) with predefined actions in order to finish the experiments started earlier in the classroom. In this application, it must be possible to see a live stream video of the work station.

Since traditionally students in a class are divided into small groups, the system must work equally for the experiments of several groups of students. For this purpose, equal sections for each group at the base of the robotic manipulator should be created. Each student should be able to use the specific credentials given by the teacher to access the corresponding area to their group.

Finally, some security needs to be implemented as there cannot be simultaneous people using the robotic manipulator. Also, while a specific group can only work in the area of their experiment, a teacher should be able to access all of the areas and interrupt a student from working at any time. To overcome some of this problems, a schedule needs to be implemented. This way, the teacher can input which group will access at what time and later only the authorized group can enter at the scheduled time.

Later, it was also decided that the system is to be called "FEUP ChemLab".

The functionalities mentioned above were divided into requirements, that is, a set of verifiable statements to evaluate the adequacy of the system to the problem.

3.1.1 Robotic Manipulator Requirements

A.1 Ability for the robotic manipulator to perform pre-defined movements through commands in the interface.

A.2 Ability for the end effector to rotate in order to transfer liquids from one recipient to another.

A.3 Ability for the end effector to hold and use a pipette.

A.4 Ability for the robotic manipulator to pick up and place chosen items in the defined position.

A.5 Ability for the robotic manipulator to transfer liquids from one recipient to another.

A.6 Ability for the robotic manipulator to pipette liquids from one recipient to another.

3.1.2 Web Interface Requirements

B.1 Be simple enough so users can use it without any extra training.

B.2 Compatible with most computers and phones without any complex installation for the user.

B.3 Allow for a user to view the workstation without any control over the robotic manipulator.

B.4 Allow only one user to control the robotic manipulator at the time.

B.5 Refuse students to enter the control page if their group was not schedule for the present time.

B.6 Redirect a user to a waiting page if it's their schedule and someone is already in the control page.

B.7 Redirect a user that's in the waiting page back to the control page when the the previous user exits the control page.

B.8 Allow teachers to be logged in at any time to supervise and kick students.

B.9 Associate a user to a group and that group to a schedule and an area.

B.10 Ability to work in delimited areas according to the user's group.

B.11 Restrict unauthorized users to other areas.

B.12 Ability for the teacher to create, change or delete a schedule.

B.13 Ability for the teacher to sign up and delete users.

B.14 Ability for the teacher to see a list of all the users and their details.

B.15 Ability for a student to see the details of their group (such as the other elements and the schedule).

B.16 Ability for a user to change their password.

B.17 Present live video stream from the workstation in the web interface.

3.2 Architecture

In the figure 3.1, the final architecture, suited to fulfill the requirements, with all the selected components is represented.

Even though, in the previous chapter, the technologies presented as a solution for processing tasks and hosting the webserver and the database are a NUC and a Raspberry, the system ended up being developed in a traditional desktop computer owned by the faculty. This decision was made mostly for convenience, as that computer was already at the work station and it didn't add any extra costs. This being said, the system can be implemented in any computer available at a school or in a inexpensive Raspberry Pi class computer.

The unmodified uStepper Robotic Arm has three motors each with its uStepper boards to be controlled trough. The gripper of the robotic manipulator is controlled using an Arduino. All four microcontrollers are connected into an USB port of the laboratory's computer.

After some initial communication tests, it was decided to create a communication platform. This decision was made to prevent the web server from opening and closing the serial port every time a message is sent. Every time the serial port opens, the microcontroller restarts and can take around two seconds to establish the communication, delaying the process. Restarting the microcontroller would repeat any movements made by the robotic manipulator initially in the setup.

The database exists to keep the user's information and control accesses.

A camera is connected to the computer so a video stream is sent to the web application.

Each part of the architecture is further discussed in the next chapters.

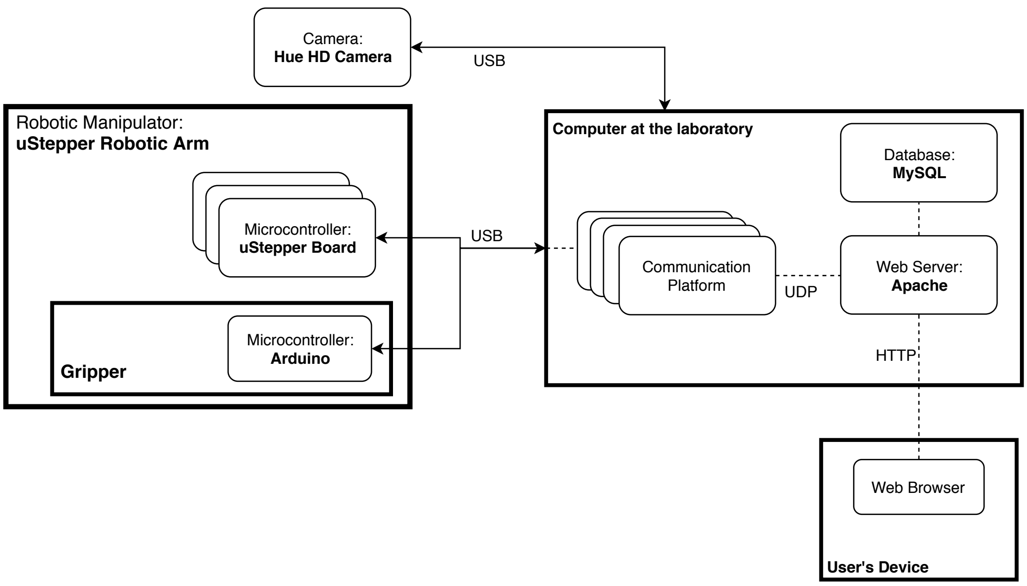


Figure 3.1: Final Architecture

Chapter 4

Web Interface

4.1 Website

A website was chosen to be the interface for this system. The main reason behind this choice is due to the fact that a website doesn't require any client side installation on the user's device and it can be open on computers, tablets or phones.

This website was built using Hyper Text Markup Language (HTML), Hypertext Preprocessor (PHP), Cascade Styling Sheets (CSS), some JavaScript (JS) and the Bootstrap library. It was developed locally on one of the Faculty's computer based on the LAMP model represented on the figure 4.1, this is, using Linux as the operating system, Apache 2.4.18 as the web server, MySQL as the database and PHP as the scripting language.

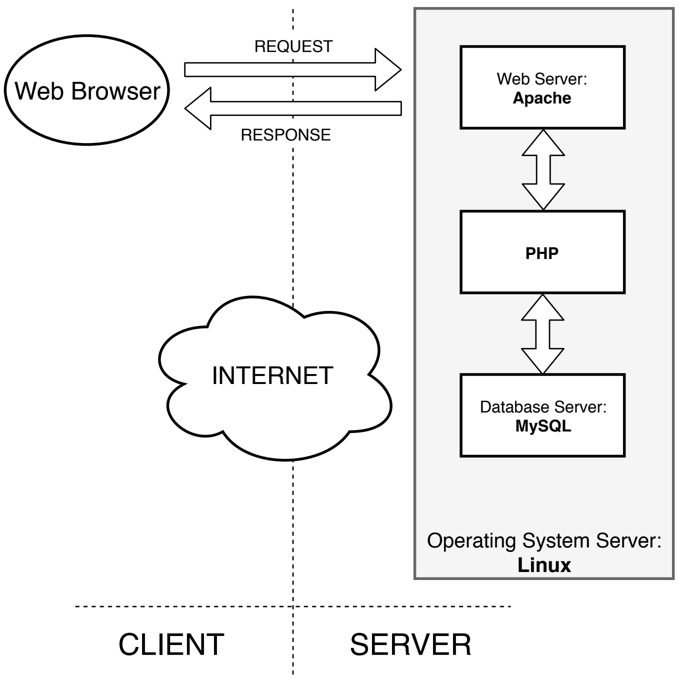


Figure 4.1: Representation of the LAMP model.

HTML is the base language for most websites. It defines the structure and presentation of raw text in a website, this is, where to position content, such as images, text, or video, defining paragraphs, headings, and data tables.

PHP, used for web development, is a server-side language which means that the code is executed on the server, generating HTML that it's sent to the client. For PHP scripts to work it's necessary to have a PHP parser (CGI or server module), a web server and a web browser [30]. The web server, with a connected PHP installation, is ran so a PHP program can be accessed using a web browser, viewing the PHP page through the server [30]. PHP supports most web servers and most databases, such as Apache and MySQL which are the used ones in this project. The version used is PHP 7.0.

JavaScript is mostly used as client-side language that determines the behavior of a website, enabling a developer to create dynamically updating content, control multimedia, animate images... In this website, JS is mostly used for dynamic content and timing events.

Bootstrap is an open source front-end framework for developing with HTML, CSS, and JS that helps building user interface components for responsive and mobile-first sites. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Using Bootstrap is very beneficial seeing that it's not necessary to write code from scratch but rather modify the existing templates and classes in a way that fits the needs of the website. Furthermore, a fluid grid layout that dynamically adjusts to the proper screen resolution, allowing the website to be correctly seen in various sized devices, is available to use with Bootstrap.

There are four main pages: home page, profile page, view mode page and control mode page. All pages have the header and the footer in common. In the figure 4.2, an organizational representation of the website's pages is shown.

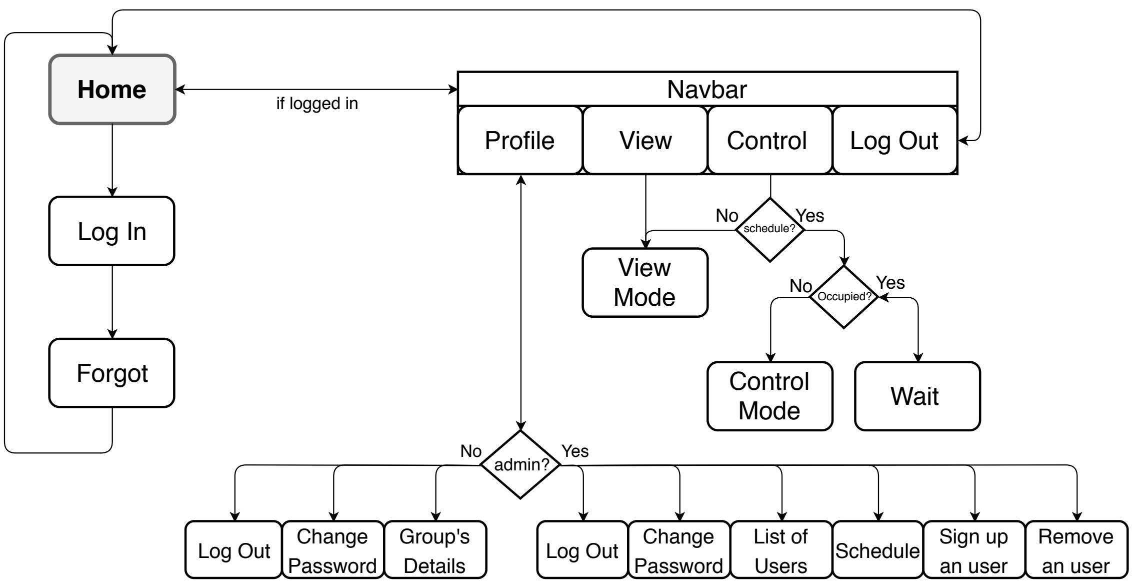


Figure 4.2: Website's map.

- **Header**

  The header sits on top of each page and it comprises a container with the website's title "FEUP ChemLab: Control of The Robotic Manipulator" and the navigation bar. It remains almost always the same in all pages helping the user navigate through the site with links to the most important pages. Visually, the differences appear on the right side of the navigation bar: if a user is logged in, only the option to log out appears and if the user hasn't logged in yet, the only option available is to log in.

  As the header is present in all pages, the same script was used to implemented JS functions and a timer to automatically log out a user if he stays inactive for more than fifteen minutes. This is, if the user doesn't click on anything or moves the mouse. When someone is logged in, a timer is initialized at 900000 miliseconds (15 minutes). When it reaches zero, the user is automatically redirected to the log out page. However, if the user just moves the mouse around or performs any action on the page the timer will restart.

- **Footer**

  The footer is the last component of the page, as it sits on the bottom of it. It's equal across all the pages and it contains a copyright and an email in case a user needs help or has any questions.

4.1.1 Home Page

The home page, shown in the figure 4.3, is simply the first page a user sees when entering the address and doesn't need any authentication to access. From there a user can log in to enter any other page.

- **Log In**, shown in the figure 4.4

  A user logs in using their email and password. Once a user successfully logs in, the website redirects to the user's profile. Besides that, a variable $_SESSION is created and the column "loggedin" from the table "users" in database is updated with the number 1 as well as the column "lastlogin" from the same table is updated with the current time. If by any reason the connection to the database failed, an alert pop ups and the user is advised to try again. Furthermore, if a user inputs an email that doesn't exist in the database or a wrong password an warning pops up and the user must try again.

  A $_SESSION variable is PHP global variable that stores information that can be used across multiple pages. This variable is used to know which user is present so that in other pages such as the profile page, information regarding this user can be fetched. Usually, this variables last until the user closes the browser or until a log out is performed.

  In case a user forgets his password, a recovery method was implemented. When the "forgot my password" button is clicked, the user is redirected to a recovery page, shown in the figure 4.5, where it's necessary to fill the email and submit. Soon after, the user will receive an

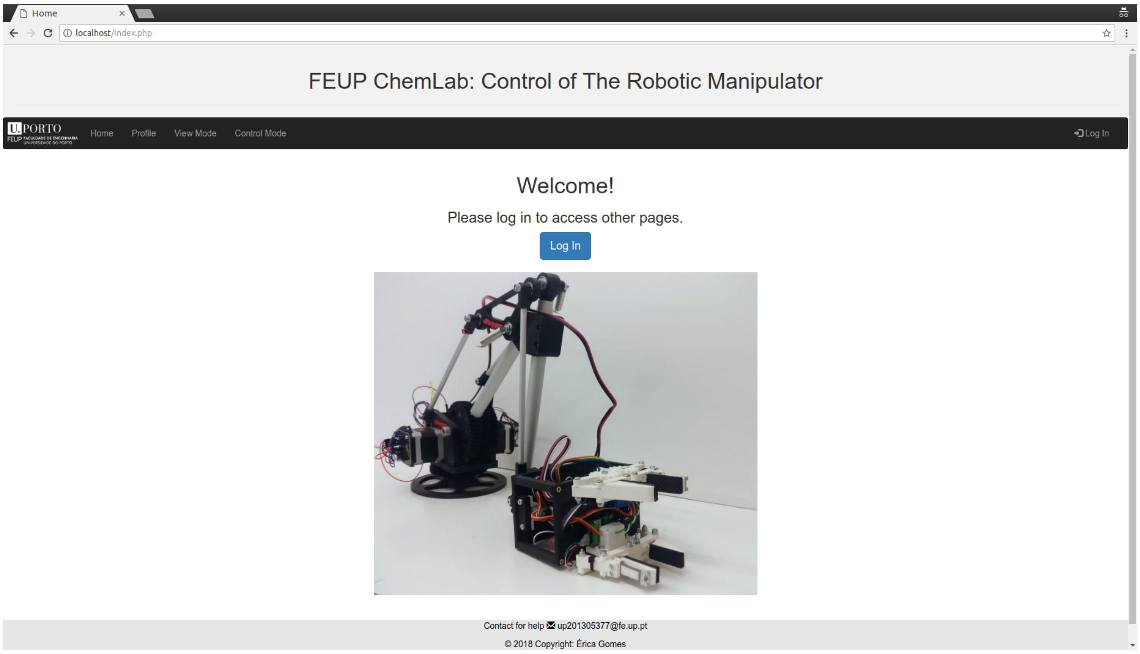


Figure 4.3: Home page.

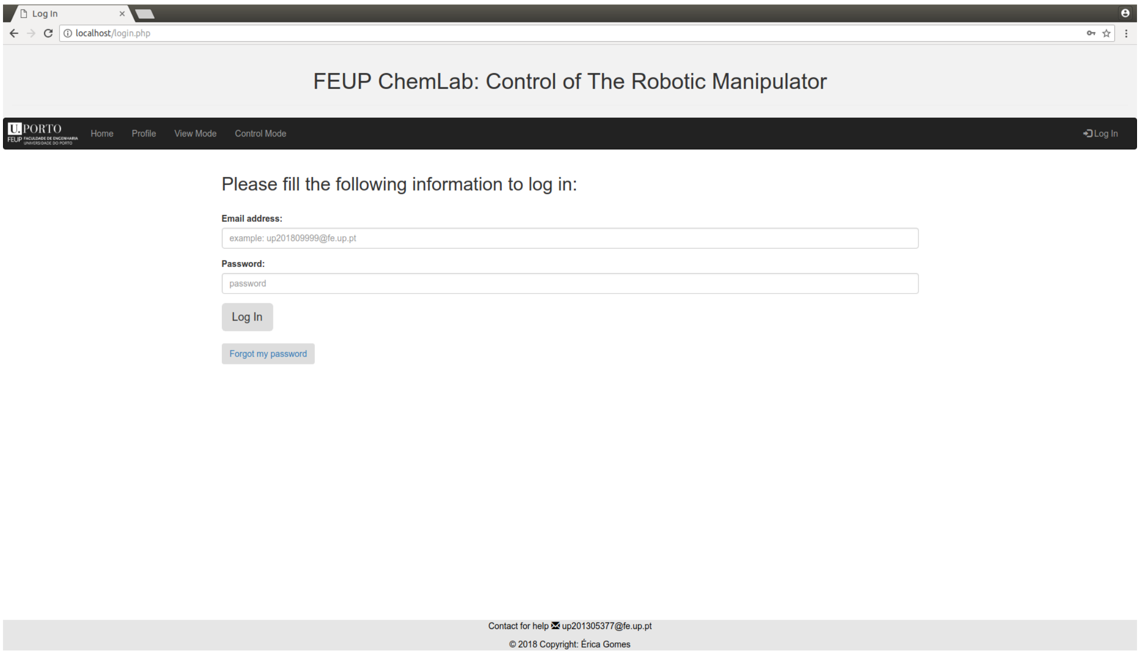


Figure 4.4: Log In page.

email from recovery@chemlab.pt with a new password (randomly generated), similar to the one shown in the figure 4.6. The password can later be changed in the profile. In order for this to work, Postfix (a mail transfer agent) was used to route and deliver the email. In other words, Postfix takes an email from the Web Server and delivers it to the specified address (email) but doesn't allow the reception of any emails.

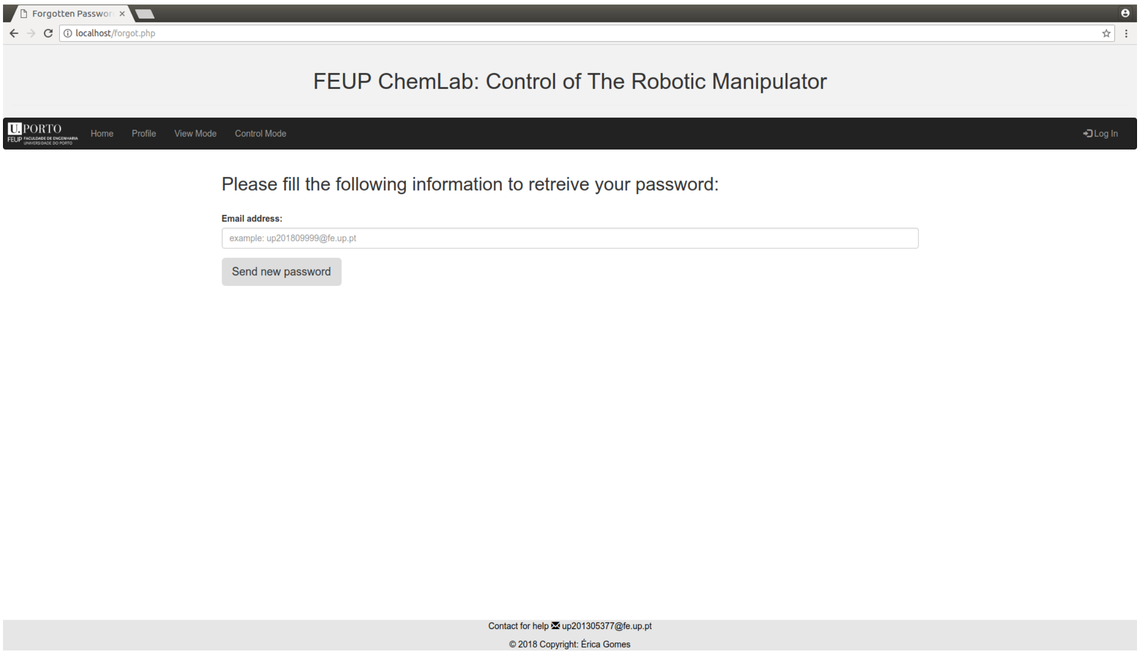


Figure 4.5: Password recovery page.

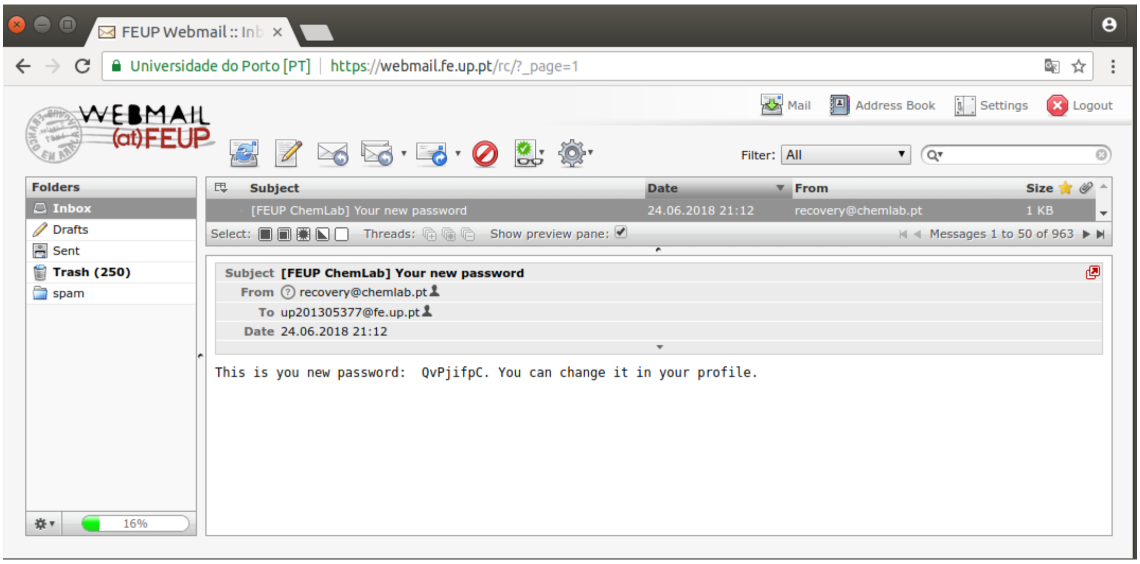


Figure 4.6: Password recovery email.

4.1.2 Profile Page

The profile page, shown in the figure 4.7, can only be accessed if a user has logged in, otherwise he will be redirected to the login page. The number of actions available varies based on the type of user: student or teacher. A teacher has access to more actions:

- **Sign up a user**, shown in the figure 4.8

  A teacher can add other teachers or students. It's necessary to complete the fields of first

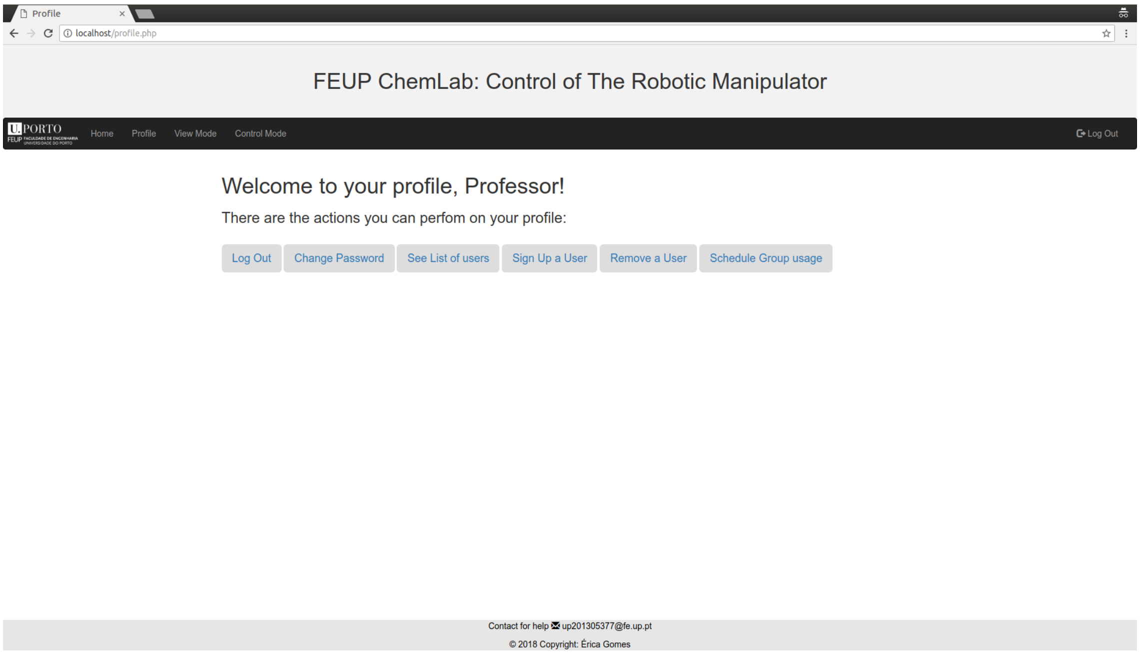


Figure 4.7: Administrator's profile page.

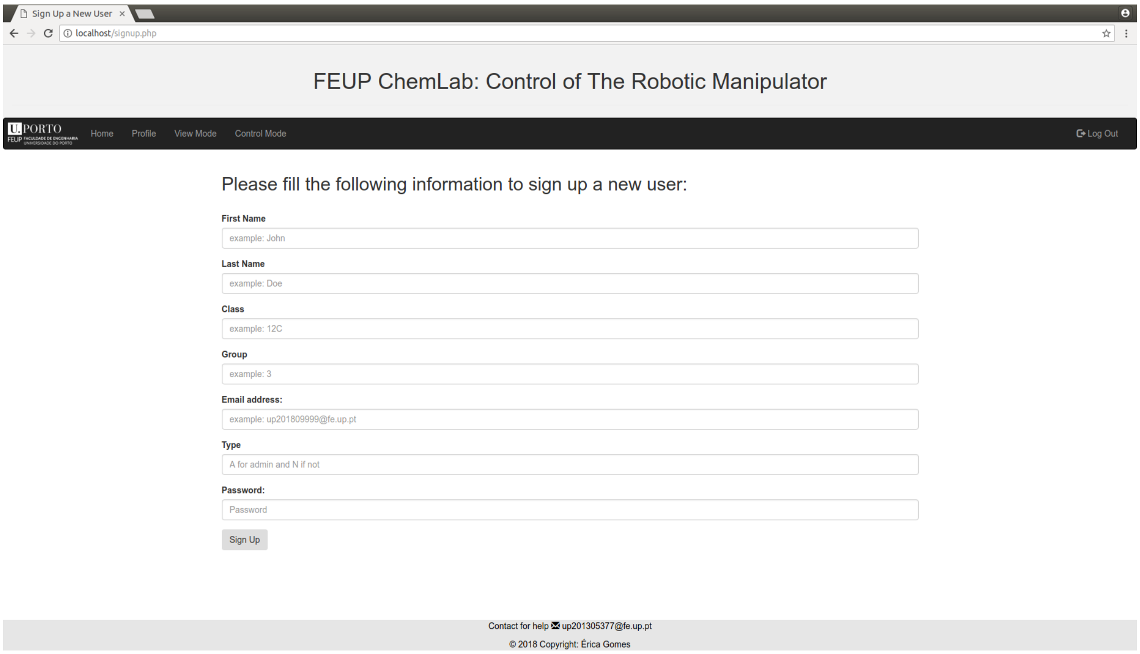
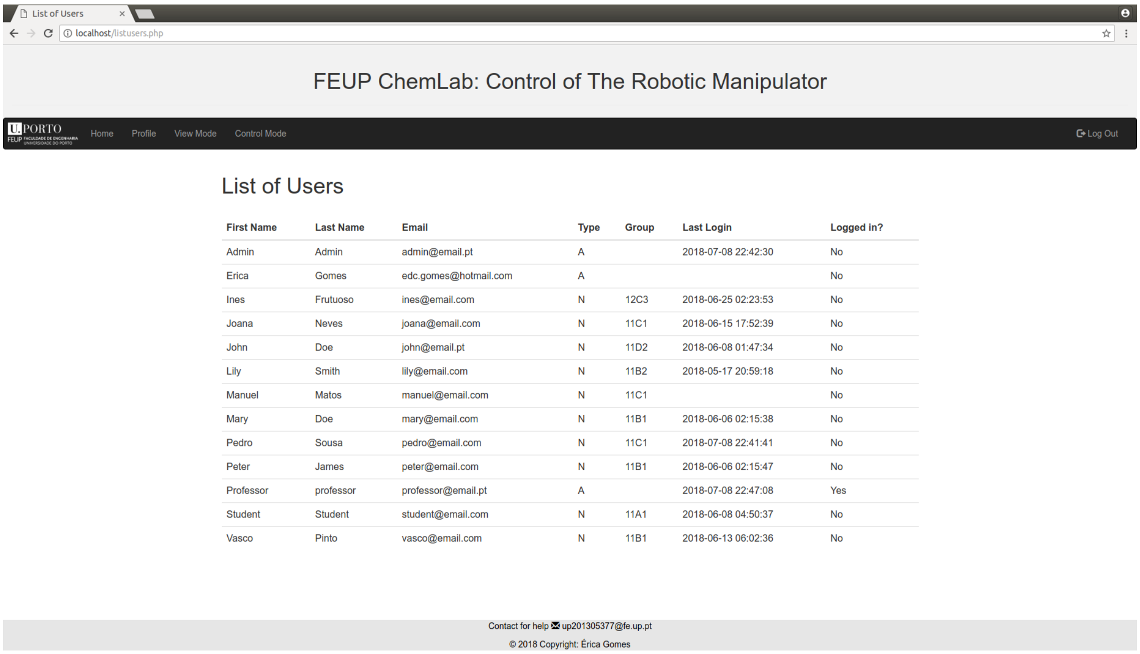


Figure 4.8: Sign up a user page.

name, last name, email, type (administrator or not) and password for all users and additionally complete the class and group for the students. If any of the required fields aren't fulfilled, an invalid email is inserted or any of the names have an invalid character (this is, not all the characters are letters), a warning is triggered and the user isn't added. Once everything is successfully filled, the information is added to the database. It should be notice that is not the password that is added to the database, but instead the hashed password is added. The password is hashed to increase protection so if by any chance anyone gets to read the database it's not possible to know the user's passwords. To hash the password, the PHP function "password_hash()" with the bcrypt algorithm is used [30].

- **See a list of all users**, shown in the figure 4.9

  This page shows a register of all users with their first name, last name, email, type, class and group, last login and if they are logged in at the present moment in alphabetic order by the first name.



First Name	Last Name	Email	Type	Group	Last Login	Logged in?
Admin	Admin	admin@email.pt	A		2018-07-08 22:42:30	No
Erica	Gomes	edc.gomes@hotmail.com	A			No
Ines	Frutuoso	ines@email.com	N	12C3	2018-06-25 02:23:53	No
Joana	Neves	joana@email.com	N	11C1	2018-06-15 17:52:39	No
John	Doe	john@email.pt	N	11D2	2018-06-08 01:47:34	No
Lily	Smith	lily@email.com	N	11B2	2018-05-17 20:59:18	No
Manuel	Matos	manuel@email.com	N	11C1		No
Mary	Doe	mary@email.com	N	11B1	2018-06-06 02:15:38	No
Pedro	Sousa	pedro@email.com	N	11C1	2018-07-08 22:41:41	No
Peter	James	peter@email.com	N	11B1	2018-06-06 02:15:47	No
Professor	professor	professor@email.pt	A		2018-07-08 22:47:08	Yes
Student	Student	student@email.com	N	11A1	2018-06-08 04:50:37	No
Vasco	Pinto	vasco@email.com	N	11B1	2018-06-13 06:02:36	No

Figure 4.9: List of users page.

- **Change password**, shown in the figure 4.10

  In this page, the user can change their own password. It's necessary to input the present password and input the new password twice. If the present password isn't correct or the new passwords don't match, a warning pops up and the user can try again. Once everything is correctly filled, the database is updated. As explained before, the password is only inserted into the database after it's hashed.

- **Schedule Group's Usage**, shown in the figure 4.11

  For a student to access the control mode page, the teacher has to schedule a time for each group which is made in this page. The teacher must fill the group ID, the initial time and the end time (day, month, year, hours and minutes). The zone, another information to fill, is the area of the workstation that the group has access and is optional as it may not have any restriction. Once the information is all filled, the database and the table bellow the form in page is updated. This table shows the current schedule (start and end time) for accessing the

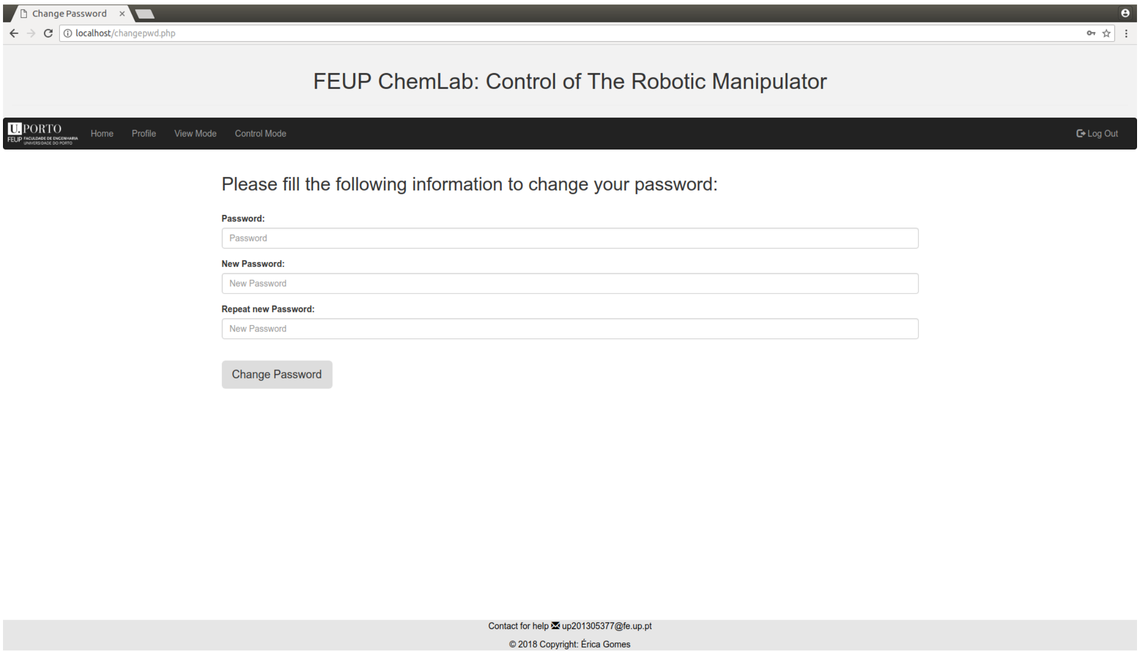


Figure 4.10: Change password page.

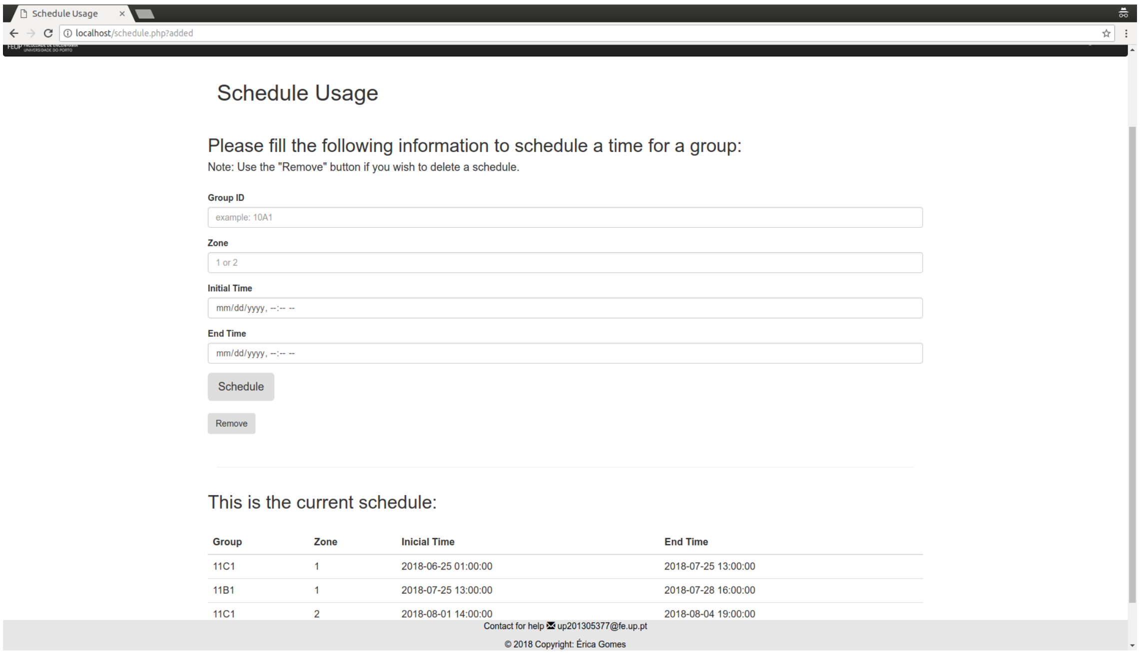


Figure 4.11: Schedule and list of schedules page.

control mode page per group as well as the zone. A warning asking for the teacher to try again will appear if the group ID filled in isn't associated with any student or the scheduled time doesn't make sense, this is the end time has already pass or it's before the start time. It's also possible for a group to have more than one schedule.

To delete a group's schedule, the teacher should repeat process explained before but click on the "remove" button instead. If the teacher tries to delete a group that doesn't exist, an alert will appear and the teacher should retry.

In order to prevent an accumulation of old schedules, every time a teacher enters this page it's checked on the database if there is any end time that is earlier than the present time, in other words, the schedule has completely pass. In that case, the so called old schedules are deleted.

- **Remove users**, shown in the figure 4.12

  A teacher can choose to remove individual users using their email or a full group of students by filling the information in the correct space. Once the "remove" button is clicked, the database is updated an the user or users disappears.

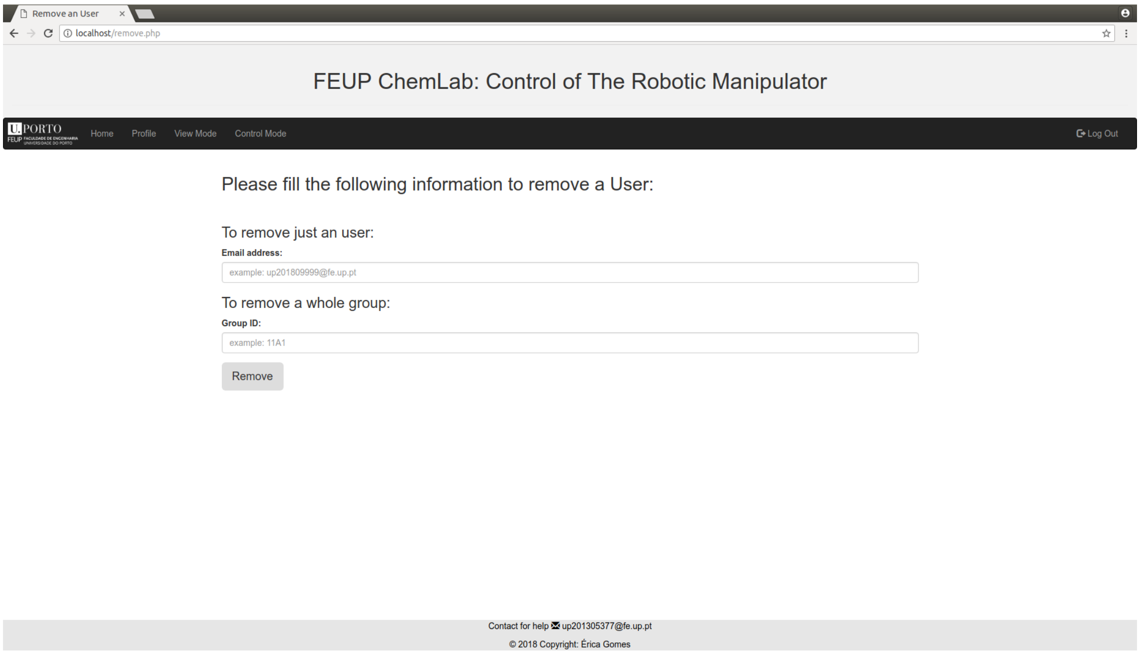


Figure 4.12: Remove an user page.

- **Log out**

  When a user wishes to log out it's only necessary to click on a "log out" button, the user will be redirected to a page confirming he has log out, shown in the figure 4.13. The session created at the time of the log in is destroyed and the column "loggedin" from the table "users" in database is updated with the number 0.

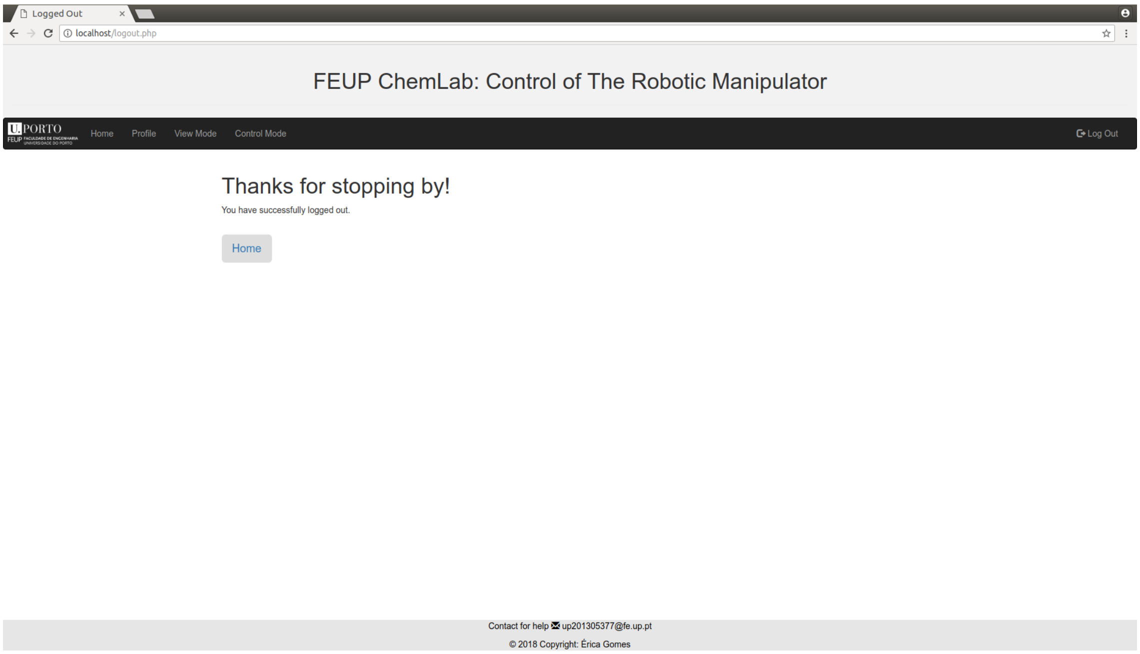


Figure 4.13: Page the user is redirect to after a log out is successful.

The student has a different profile page, shown in the figure 4.14, and can perform the following actions:

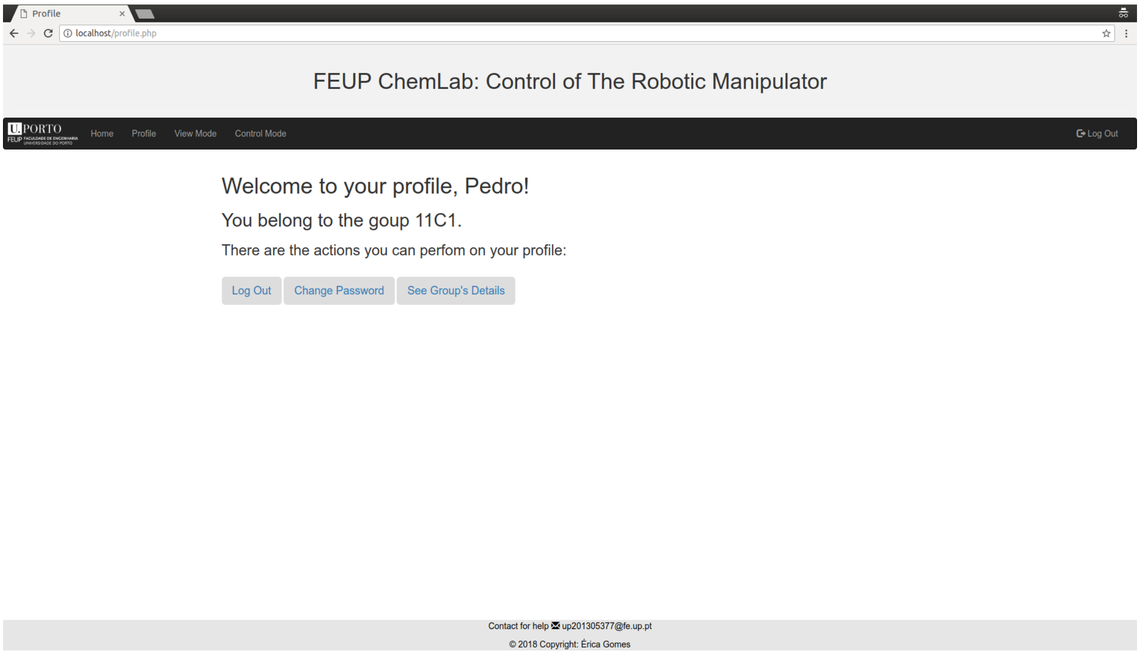


Figure 4.14: Student's profile page.

- **Change password**

  The proceedings in this page are the same as the ones for the "change password" page for the teachers, shown in the figure 4.10 and explained previously.

- **Logout**

  The proceedings in this page are the same as the ones for the "Log out" page for the teachers, explained previously and shown in the figure 4.13.

- **See group's details**, shown in the figure 4.15

  In this page, each student can see the schedule for their own group as well as the list of first names, last names, emails, last log in and if the elements of their group are on-line.

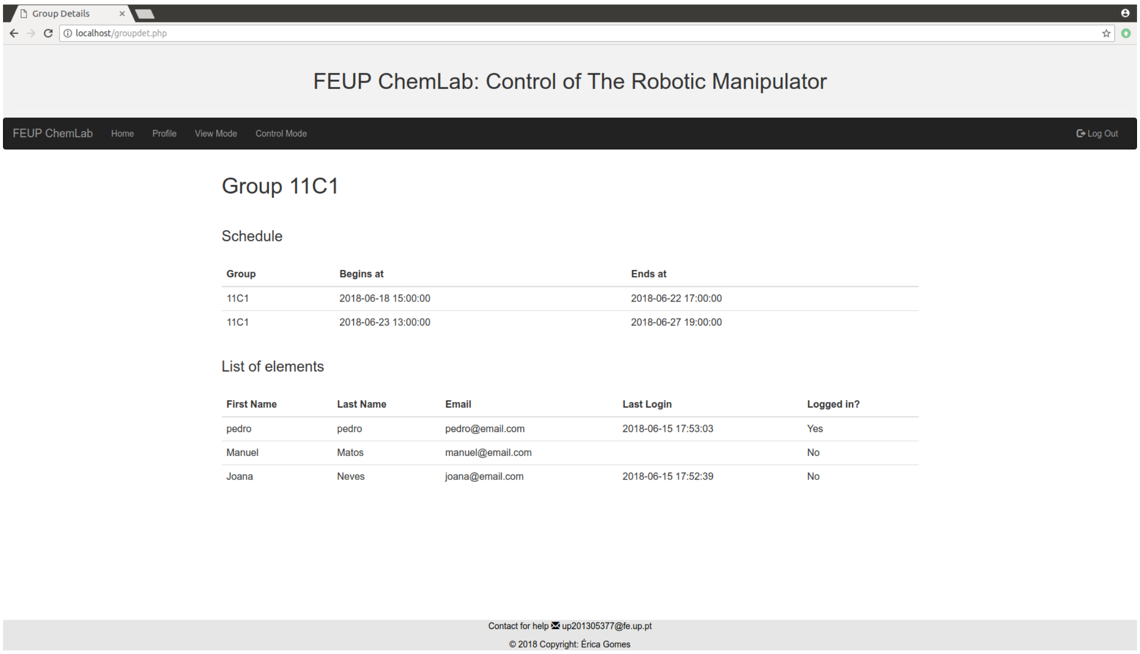


Figure 4.15: Group's details page.

4.1.3 View Mode Page

The view mode page, shown in the figure 4.16, can be accessed by anyone who is logged in. It shows a live video feed of the laboratory work station, with the camera pointed at the robotic manipulator. In this figure, a workstation with a pipette on a 3D printed pipette support, a petri dish, a beaker and a test tube on a 3D printed test tube support can be seen.

The VLC Media Player is used to stream the video from the Hue HD camera to the local address http://192.168.105.71:8080/stream.ogg which is the source of the video presented in the view page.

The video image that is presented in the website has a delay of around five seconds. To measured this delay, an outside camera filming both the robotic manipulator and the website was placed during a test experiment. The time interval between the moment the action starts on the outside camera and the moment the action starts on the live feed was checked several times.

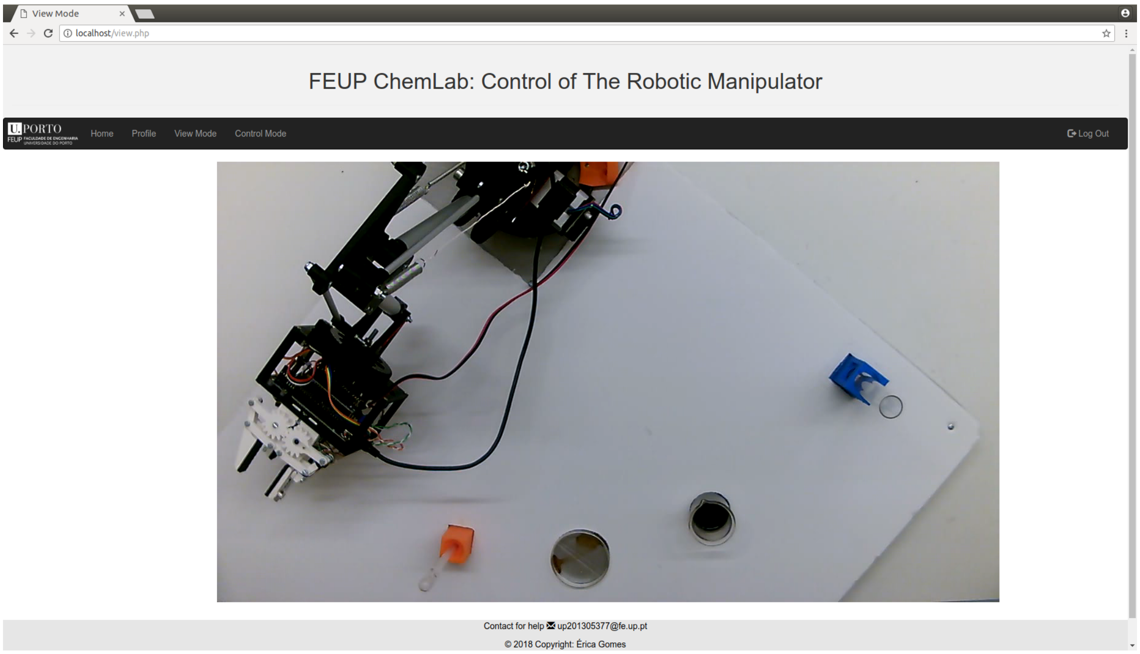


Figure 4.16: View mode page.

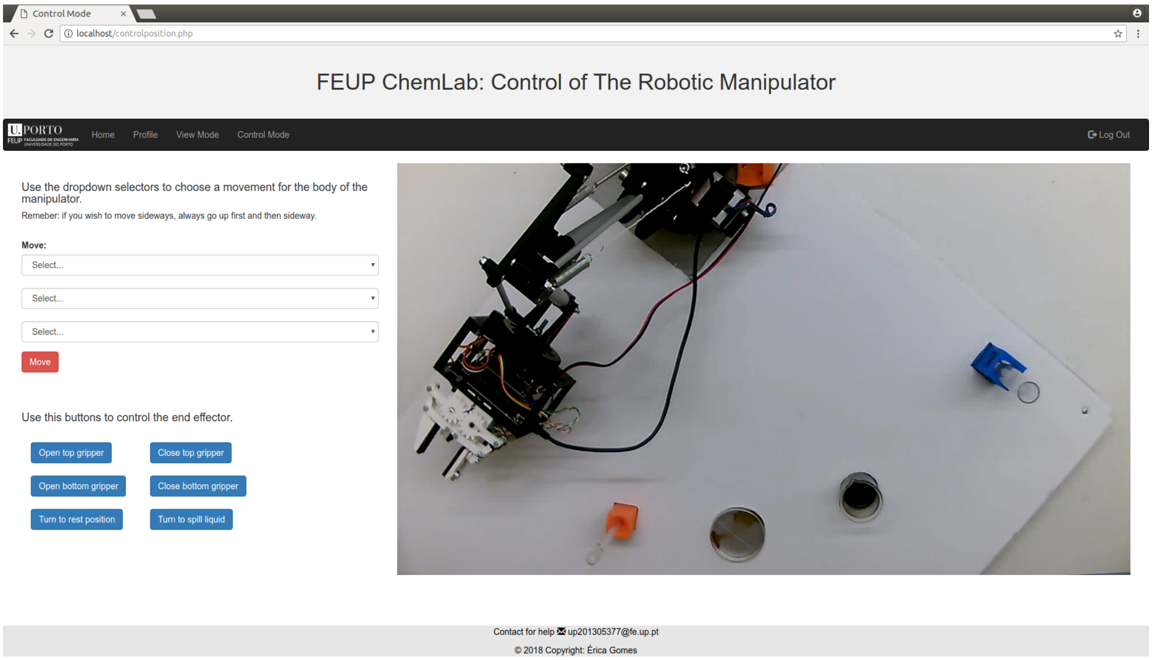


Figure 4.17: Control mode page.

4.1.4 Control Mode Page

The control mode page, shown in the figure 4.17, can only be accessed by users who are logged in and, in case of a student, within the assigned schedule. It allows a user to move the robotic manipulator to predefined positions using buttons available and seeing the progress by a video in the same way as in the view mode page. In the figure 4.18, a sequence diagram of the actions performed by each part of the system when a movement is executed is presented.

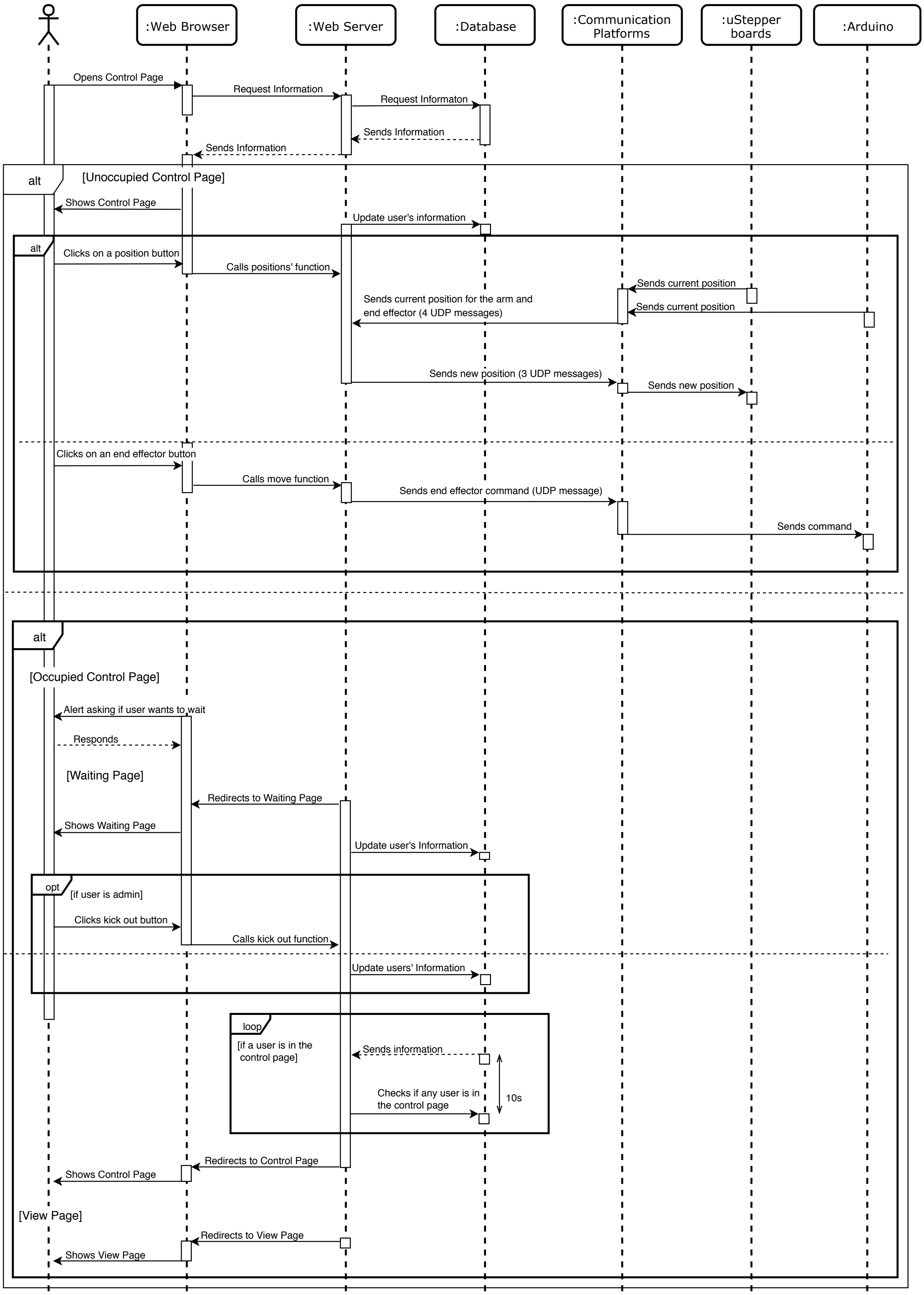


Figure 4.18: Sequence diagram for the control page.

When a user first opens the control page, several informations that are relevant for the good functioning and the appearance of the page are checked:

– Occupancy of the page: it's checked if a user is already controlling the robotic manipulator in the database. If the page is occupied an alert is shown to the user and the option to go to a waiting page is given. The waiting page is discussed later on in this section. If the user doesn't choose to go to the waiting page, he is redirected to the view mode page.

– Type of user: if the user is an administrator, in other words a teacher, the schedule will not be checked as it doesn't exist.

– Schedule: if the student's group was schedule for the present time, the database is updated indicating that a user is present at the control mode page. If the group's schedule has already pass an alert appears informing the student of such and that he should ask the teacher for a new schedule. In the other hand, if the schedule hasn't passed yet and is not at the present moment an alert surfaces notifying the student that the schedule is at a later time and he can check his profile to confirm.

– Zone: The buttons available for each student depend on what zone or zones they can access, as indicated by the teacher at the moment of the schedule. A teacher can access all zones at any time as long as the control page is not being used.

Once everything is checked, the control page shows up and the user is able to control the robotic manipulator. However, as a safety precaution, the user is not able to freely control the manipulator and can only move it to predefined positions. This is also beneficial for the user as it's simpler to use.

Since the positions for the robotic manipulator are predefined, they are labeled according to a code composed by a letter and a number. The number corresponds to an object and the letter to a position relative to the object.

For the setup shown in the figure 4.17, with a pipette on a 3D printed pipette support, a petri dish, a beaker and a test tube on a 3D printed test tube support (from left to right of the image), the positions are coded using the table 4.1.

Table 4.1: Meaning of each letter and number of the position code.

Meaning	**Letter**	**Meaning**	**Number**
over the	A	Pipette	1
on the	B	Test Tube	2
position to collect liquid from	C	Beaker	3
position to drop liquid into	D	Petri dish	4
over the position to transfer liquid to	E		
position to transfer liquid to	F		

An action of the robotic manipulator can be defined as the moving of the object held by the dual gripper (or solely the dual gripper if no object is held) to a position.

The user can indicate the action to perform by filling the form on the left top side of the control mode page, creating a sentence describing the action by choosing from predefined options shown in the table 4.2. The user can also click on one of the end effector's buttons (presented in the bottom left side of the control page) to send the command described on the button.

Table 4.2: The three sets of the form options to make a sentence.

First Space	Code	Second Space	Code	Third Space	Code
Move pipette	1	to over the	A	pipette	1
Move test tube	2	to the	B	test tube	2
Move free gripper	0	to collect liquid from	C	beaker	3
		to drop liquid into	D	petri dish	4
		to over the position to transfer liquid to	E	pipette support	1
		to the position to transfer liquid to	F	tube support	2

For safety reasons, the action submitted by the user is analyzed before being performed. Before the analyzes, the web server receives the angle of each part of the arm and the end effector through User Datagram Protocol (UDP) messages from the communication platform. The analyzes consists of three steps.

– First, the current angle of the end effector is checked, if it's not in its resting position (near zero degrees) the user can't perform a movement and is warned to move the end effector to a rest position.

– Secondly, the current and the final position and compared on a logical level. This is, if it makes sense to perform that movement, even if the movement has no physical restrictions. If the movement makes no sense, a warning is sent to the user and the movement isn't performed, similarly to the previous check. One example of this is to move a test tube above the pipette support, it's possible but it makes no sense to do so in an experiment as you can't transfer liquid from a test tube to a pipette. The figure 4.19 shows all the logical actions for the considered case.

– Finally, the current position of the body manipulator is checked and it's compared to the position the user is going to send the manipulator to. If the movement is not physically possible or safe (for instance, if doing that movement means hitting objects on the way to the final position) then the user is warned and the action is not performed. The table 4.3 shows if it is physically possible and safe to reach the intended next position based on the current position. It is important to note that the absent positions on this table, for instance C1 (position to collect liquid from pipette), are positions that are not considered valid for this test case. The possible movements are represented in green.

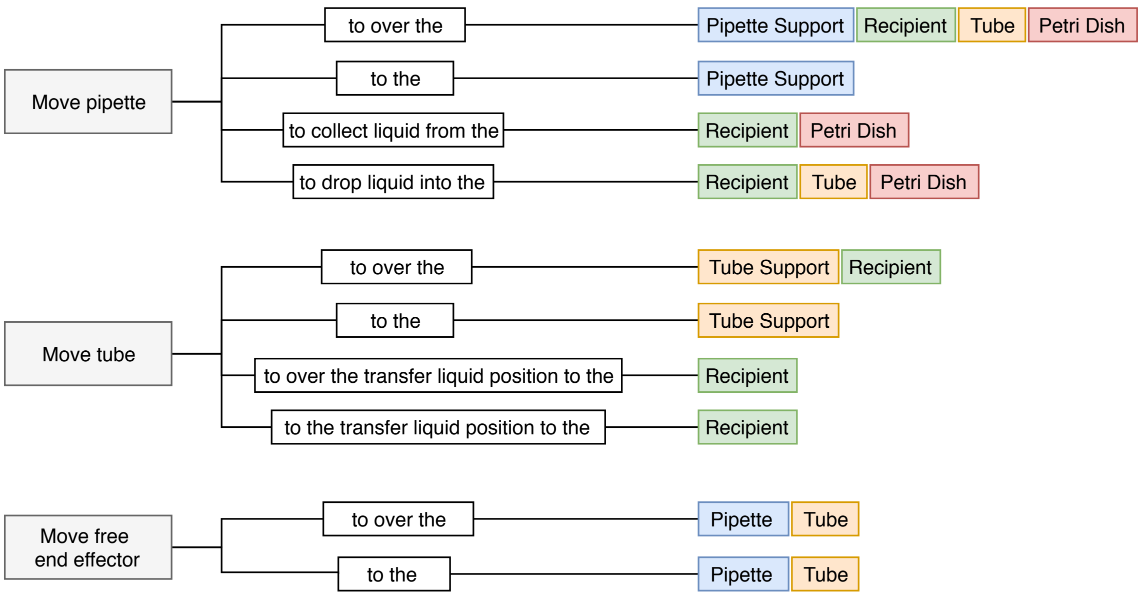


Figure 4.19: Logical actions.

Table 4.3: Physical interpretation (with the meaning of the code in the table 4.1).

Current position \ Next position	A1	B1	A2	B2	D2	A3	C3	D3	E3	F3	A4	C4	D4
A1	0	1	1	0	0	1	0	0	1	0	1	0	0
B1	1	0	0	0	0	0	0	0	0	0	0	0	0
A2	1	0	0	1	1	1	0	0	1	0	1	0	0
B2	0	0	1	0	0	0	0	0	0	0	0	0	0
D2	0	0	1	0	0	0	0	0	0	0	0	0	0
A3	1	0	1	0	0	0	1	1	1	0	1	0	0
C3	0	0	0	0	0	1	0	1	0	0	0	0	0
D3	0	0	0	0	0	1	1	0	0	0	0	0	0
E3	1	0	1	0	0	1	0	0	0	1	1	0	0
F3	0	0	0	0	0	0	0	0	1	0	0	0	0
A4	1	0	1	0	0	1	0	0	1	0	0	1	1
C4	0	0	0	0	0	0	0	0	0	0	1	0	1
D4	0	0	0	0	0	0	0	0	0	0	1	1	0

After this analysis, a JS function calls a different PHP script. This script has the position of each motor saved for the specific position associated to the code. The web server sends three UDP messages with the positions of the three motors to three different ports. These messages are received by the communication platforms which consequently sends the positions to each uStepper board.

When an end effector button is pressed, like previously, a function calls a different script. The web server then sends an UDP message with the corresponding command to the communication platform of the end effector. Unlike previously, the message sent is not a position (a single number) but a command corresponding to an end effector's action, which is further explained in the chapter 6.

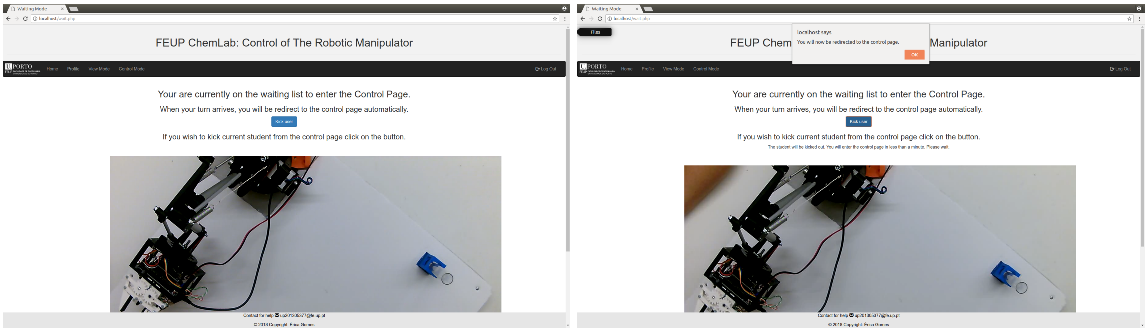
In order for a student not be able to stay in the control mode page past the scheduled time, every 10 seconds a JS function calls a script that checks on the database the end time and compares it to the current hour. In case the student has exceed the scheduled time, an alert with that information appears and the user is redirected to the view page.

When a user is redirected to the **waiting page**, the database is updated indicating that a user is waiting to enter the control mode page and with what time the user entered this mode. The page, similarly to the view mode page, shows a video feed of the what's happening at the moment in the laboratory. While one or more users are waiting, every ten seconds it's checked in the database if there is still a user in the control mode page. If there is not a user in the control page anymore, it's checked which is the user that is waiting the longest. Then, that user is redirected to the control mode page.

The waiting page for a teacher is presented in the figure 4.20a and the page for a student is the one shown in the figure 4.21.

A "kick out" feature was also implemented. In the manner that, if a student is in the control mode page and a teacher in the waiting page, the teacher can force the student out of the control page. For this, when the teacher clicks the "kick out" button, the database is updated. The same script, called every 10 seconds, that checks if the scheduled time as passed also checks, in the database, if the user needs to be kicked out. In that case, the user receives an alert with such information, as shown in the figure 4.20b, and is redirected to the view mode page.

Notice that it's extremely important to make sure that the database always has the correct location of the user, this is, that it indicates correctly if a user is in the control mode page, in the waiting page or needer of those two. For this, in every page, when a user accesses it, the database is updated with the location. Additionally, for when a user leaves the waiting page, the entry time is changed to null at every page accessed other than the waiting page.



(a) Page while waiting.

(b) Page after clicking the kick button and the control page gets unoccupied.

Figure 4.20: Waiting page for a teacher.

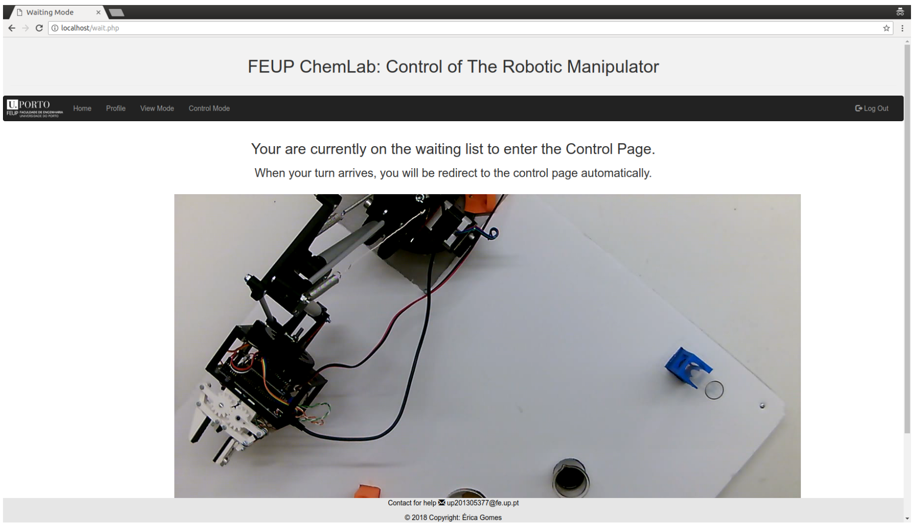


Figure 4.21: Waiting page for a student.

4.2 Database

A database is needed to keep information about the users and control the accesses. As shown in the figure 4.22, two tables were created: one with the users' information and another for the groups' information. These tables are related via the foreign key "group_id".

A user can belong to one or no group and a group can have one or multiple users. A group can have zero, one or multiple schedules and a schedule can be linked to one or multiple groups.

- Users Table

  The primary key is the "id" column which is automatically incremented when a new user is added.

  The "firstname" and "lastname" columns can't be null and as the name indicates it keeps the first and last name of a user.

  The "admin" column can't be null and stores the type of user, being 'A' or 'a' for administrator (a teacher) and 'N' or 'n' for not administrator (a student). The email column is a unique, not null column that saves the user's email.

  The "password" column can't be null and holds the hashed password of the user.

  The "group_id" column is a foreign key and can be null as the teachers aren't associated with any group and it keeps the class and number of the group of a student.

  The "lastlogin" column can be null considering that a user might never have logged in and is updated every time the user logs in.

  The "loggedin" column can have different values: 0 meaning the user is logged out, 1 indicating the user is logged in, 2 representing a user in the control mode page and, showing that a user is on the waiting page and 4 indicating that a teacher is kicking the student from the control mode page.

  The "wait_entrytime" column can be null if the user is not on the waiting page and it's updated with the date and time of the moment when a user enters that page.

- Schedule Table

  The primary key is the id column which is automatically incremented when a new schedule is added.

  The "group_name" column is a foreign key, it can't be null and it keeps name of the group (composed by the class and number of the group).

  The "begin" column saves the start date and time a group can access the control mode page and it cannot be null.

  The "end" column holds the deadline date and time for a group to access the control mode page and it cannot be null.

  The "zone" column indicates which is the are of the laboratory table that the robotic manipulator sits on that the group can access. It can be null when a group doesn't have restrictions.

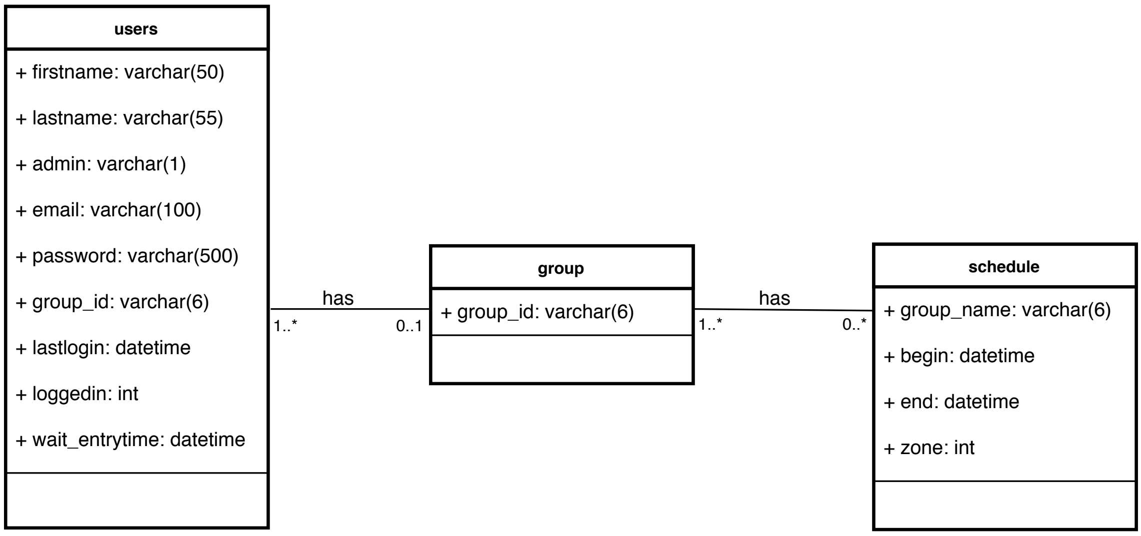


Figure 4.22: Representation of the database.

Chapter 5

Robotic Manipulator

5.1 Body of the Manipulator

A uStepper Arduino library is available in the uStepper's GitHub[22] with software functions for the uStepper board. From this library, the functions to stop the motors, to move the motor to a relative angle, to get the current angle of the motor shaft and to set the velocity and acceleration are used in the control program of the robotic manipulator.

In the figure 5.1, it's possible to see the base of the robotic manipulator with the motors, boards, the names used to identify each part and the limit switches.

Each of the uStepper boards controls one of the motors of the manipulator. The arm has two sets of two gears responsible for the tilt motion. A third set of two gears, located at the bottom of the robotic manipulator, is responsible for the pan movements, this is, the rotation of the arm. The pan movements are represented as the movements 0 and 2, in the figure 5.2, which are the responsibility of the motors 0 and 2 and tilt movements are represented as the movement 1, in the same figure, which is the responsibility of the motor 1.

As explained previously, the uStepper board is capable of knowing if the motors are moving and where they are. It continuously monitors where the motor is, where it should be and compensates if needed. This position tracking is possible due to the 12-bit rotary absolute encoder, meaning that the shaft position can be tracked in steps.

Using the function stepper.getAngle() from the class accessing all features of the uStepper board present in the library, it's possible to get the current shaft angle. This angle is the angle of the small gear, represented as α in the figure 5.3. However, the angle needed is the one from the bigger gear represented in the figure 5.3 as θ.

The arm's gears have a 4,09 to 1 ratio. This means that when the big gear from the arm completes a full turn, the small gear turns 4,09 times. As illustrated in the figure 5.4, if a read from the encoder gives the angle 120° there are four different possible positions for the big gear: 60.66°, 150.66°, 240.66° or 330.66°. Therefore, additional work is needed to solve this.

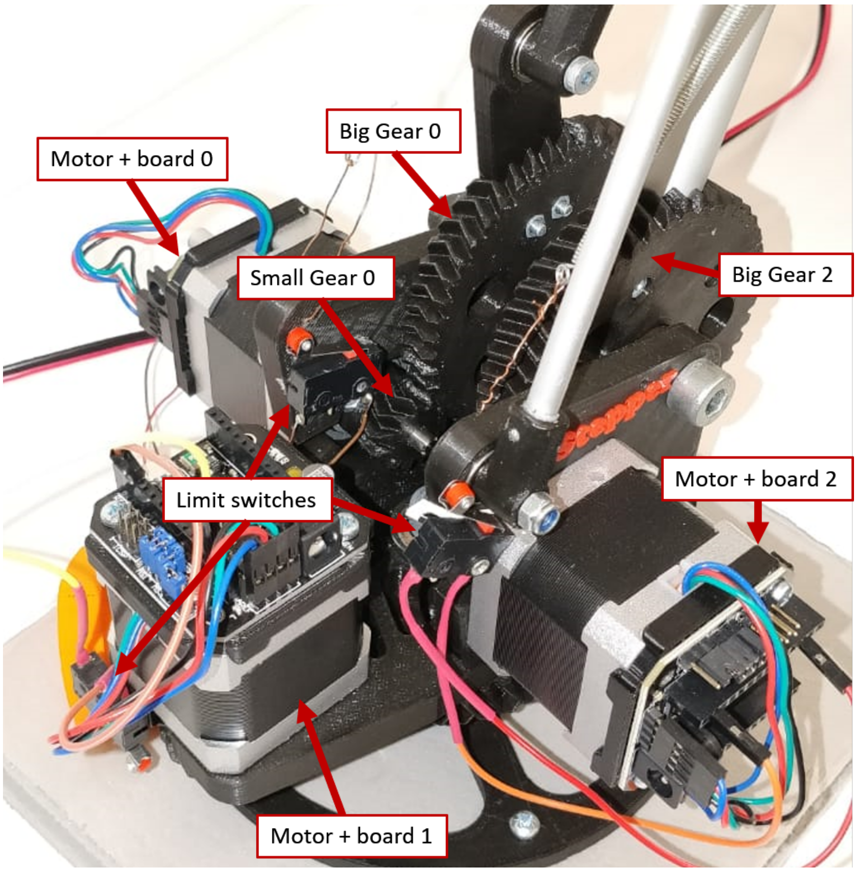


Figure 5.1: Base of the robotic manipulator with labels.

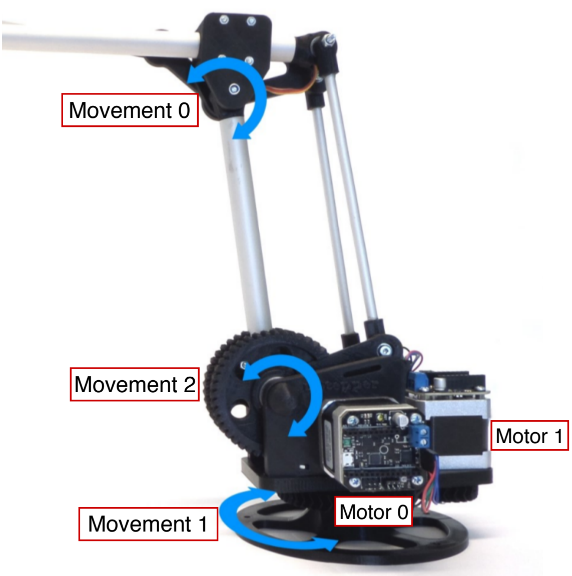


Figure 5.2: Representation of the possible movements and the responsible motors. Adapted from [21].

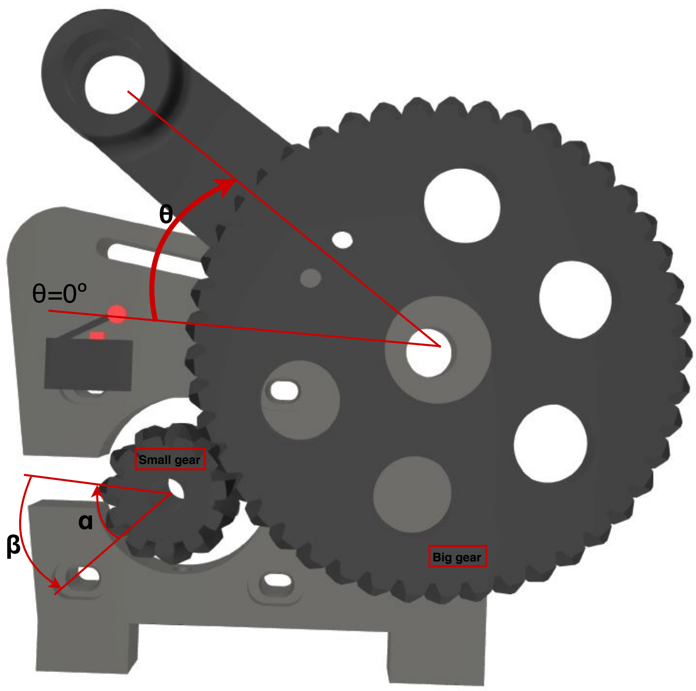


Figure 5.3: Modeled lateral view of the gears of the motor 0.

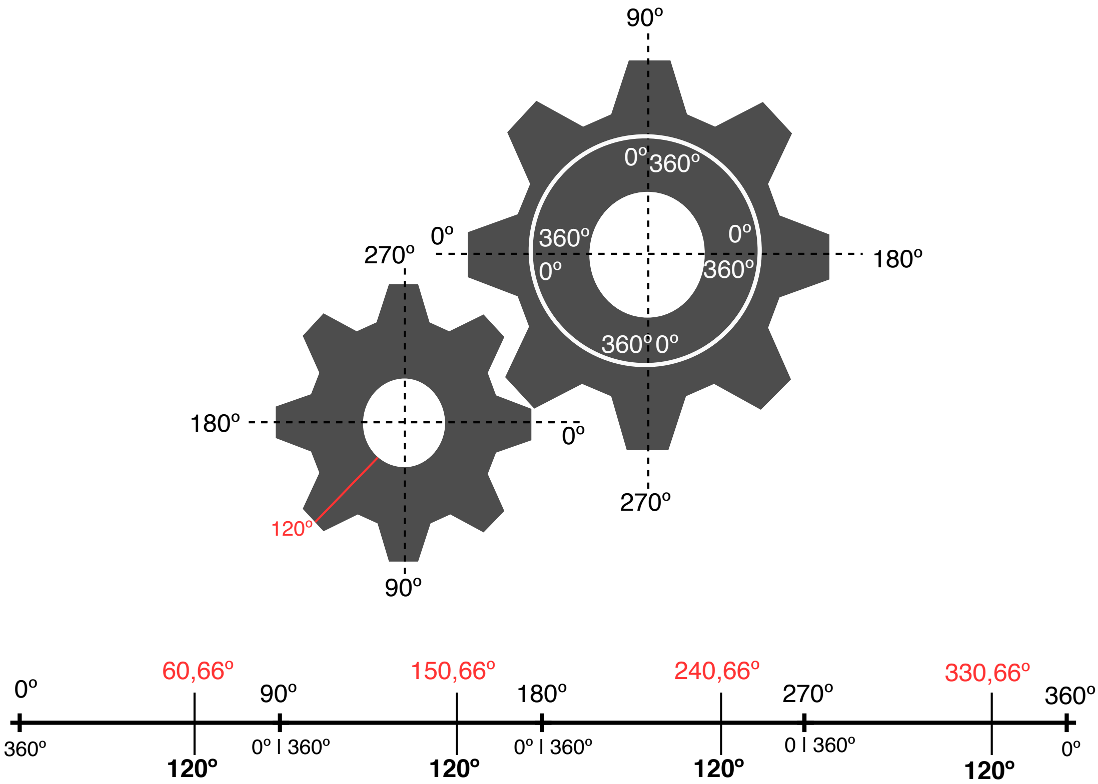


Figure 5.4: Encoder problem representation for the motor and uStepper board 0.

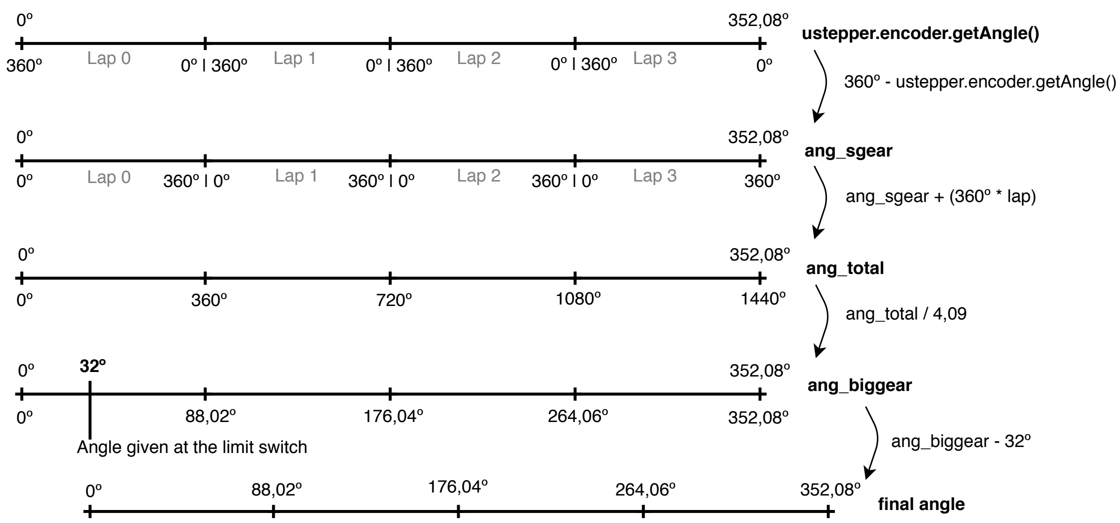


Figure 5.5: Encoder angle transformation for the motor and uStepper board 0.

Figure 5.5 illustrates the method used to track the θ angle of gear 0. As shown in the first line, for θ (shown above the line) the readings of the encoder (or α) are the ones shown below. Since the gears always rotate in opposite directions, when the value of θ increases the value of α decreases, so the angle of the small gear was reversed, as shown in the second line. This transformation corresponds to β, drawn in the figure 5.3.

Next, the concept of laps was created to discover when the small gear had made a full rotation and in which direction. The difference between the current β angle and the previous β angle is calculated in the loop cycle. This being said, if β is 360° and the previous angle is 0°, there's a difference of $-360°$ which means the angle is in the next lap. If the difference is 360° then the angle is in the previous lap.

This is crucial to transform the four intervals of angles given by the encoder into one continuous interval. For this, it's added to the given shaft angle 360° times the number of the lap. Finally, to put the angles in the same range of values, the angle is divided by the gear ratio 4,09.

A limit switch was placed next to the big gear to ensure that the angle positions saved in the website remain the same every time. When the robotic manipulator is initialized, the arm moves in the direction of the limit switch and stops when it reaches there. Every time the limit switches are activated, the number of laps resets to zero, this causes the laps to always correspond to the same angle range.

Additionally, to shift the 0° position to the where the limit switch was placed, as represented in the figure 5.3, the angle at that position was checked (with a result of 32°) and taken from the final angle, has shown in the last line of the figure 5.5.

With this transformation, that has the final equation $\theta = \frac{(lap*360°)+(360°-stepper.getAngle())}{4{,}09} - 32°$, it's possible to get the angle of the arm's big gear 0, at every instant.

The method explained above is intended for the motor and uStepper board 0, nevertheless it can be used for the gears corresponding to the motor 2 with a few alterations, as it can be seen in the figure 5.6.

There is no need to reverse to small gear angle, considering this gear rotates in the positive direction as the angle of the arm. The other modification is the different value angle removed to shift the 0° position to the where the limit switch was placed. In the same way as previously, the angle at that position was checked (with a result of 67°) and taken from the final angle. The equation $\theta = \frac{(laps*360°)+stepper.getAngle()}{4{,}09} - 67°$ is the result of the method applied to the gears of the motor 2.

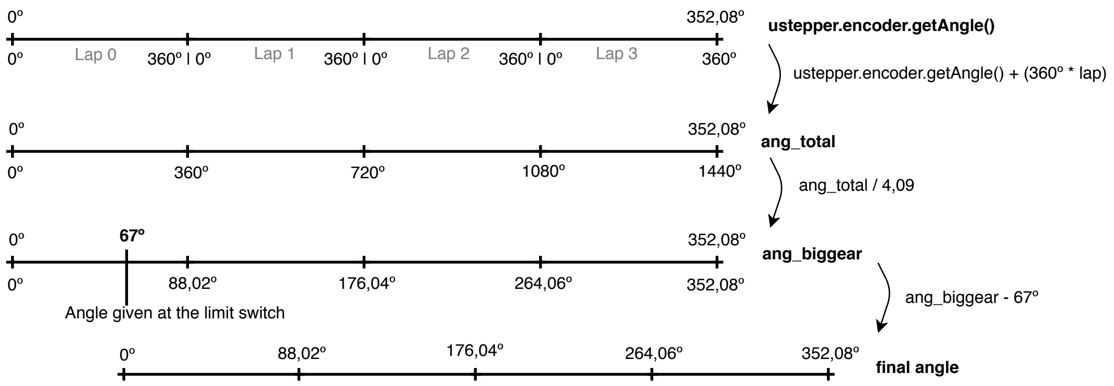


Figure 5.6: Encoder angle transformation for the motor and uStepper board 2.

The gears responsible for the pan movements have a 2 to 1 ratio, this means that when the big gear completes a full turn, the small gear turns two times. Similarly to the arm's gears, the angle desired is the one from the big gear. To obtain this, a simpler version of the method previously explained is used and represented in the figure 5.7. A limit switch was also placed to ensure that the angle positions saved in the website always correspond to the same position. A smaller gear ratio translates in fewer laps and as there's no benefit in starting in a specific position there is no need to shift the angle. Culminating in the end equation $\theta = \frac{(laps*360°)+(360°-stepper.getAngle)}{2}$.

Additionally, it's important to notice that this method prevents the arm from rotating beyond the 360°. If that was allowed, the power and USB cables would wrap around the base of the manipulator and possibly disconnect.

For the limit switch of the motor 1 to be activated, a 3D printed piece was placed in the base of robotic manipulator shown in the figure 5.8.

Once the current angle is calculated and an angle position has been received from the serial port, the difference between those angles is calculated and passed to the function stepper.moveAngle(). This function moves the motor to a relative angle from current position. A positive angle makes the motor turn clockwise, and a negative angle, counterclockwise. For the motor 0, a positive value is passed to the function when it's necessary to move the arm forward (in the positive direction of θ) and a negative one when it's needed to move the arm backwards (in the negative direction of θ). In the other hand, for the motor 2, as it's facing the motor 0, a negative value is passed to the function in order to move the arm in the positive direction (forward) and a positive value also multiplied by 4,09 moves the arm in the negative direction of θ (backwards). For the motors

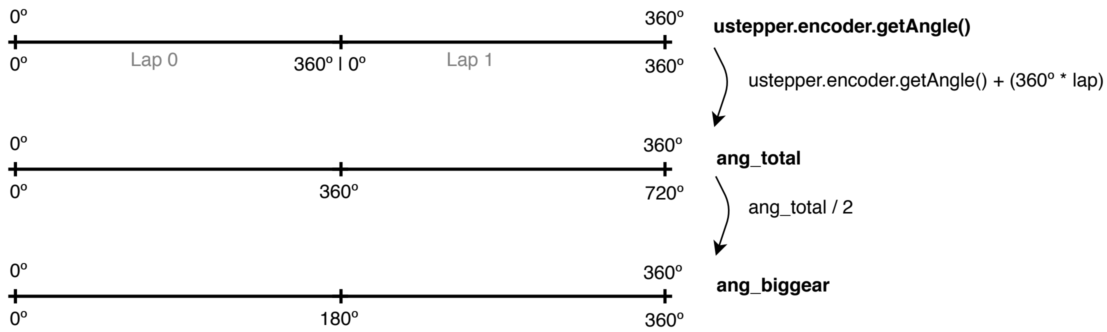


Figure 5.7: Encoder angle transformation for the motor and uStepper board 1.

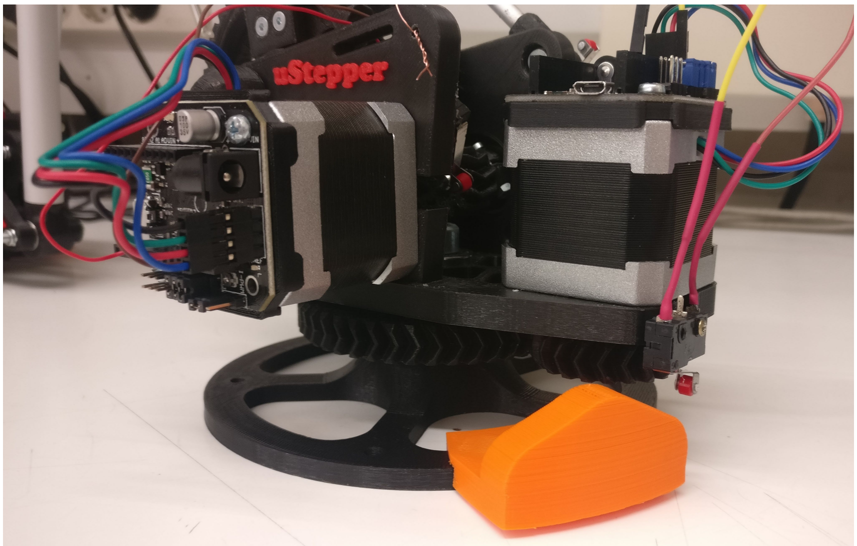


Figure 5.8: 3D printed piece and limit switch for the motor 1.

0 and 2, the value passed is multiplied by 4,09. This happens because the difference calculated corresponds to the position in the big gear and the function moves the small gear, meaning it's necessary to compensate. Regarding the motor 1, the value passed is multiplied by 2 for the same reason.

In order to prevent spillage of liquids during movements, the acceleration in $steps/s^2$ of the motors was set at a low value using the function stepper.setMaxAcceleration() from the library. The velocity in $steps/s$ can also be altered using the function stepper.setMaxVelocity().

5.2 End Effector

The original gripper, shown in the figure 5.9, can only open and close having no ability to rotate. As one of the final goals of this manipulator is to perform chemistry experiments, more complex movements are needed. Therefore, a new end effector was designed with a wrist joint in order to transfer liquids from one recipient to another and dual grippers to pipette liquids, a fundamental operation in chemistry. The dual grippers will also add stability when moving an object.

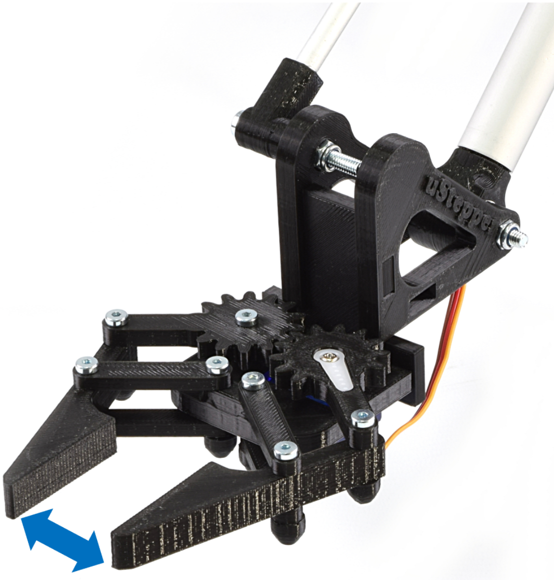


Figure 5.9: Original gripper. Adapted from [31].

The new end effector, shown in the figure 5.10 is composed by 3D printed parts from the original gripper, colored black in the figure 5.11, and some new ones, colored in red in the same figure. In the middle of the two plates an Arduino Uno, an Adafruit Motor Shield V2 and a YFRobot Sensor Shield are fitted on top of each.

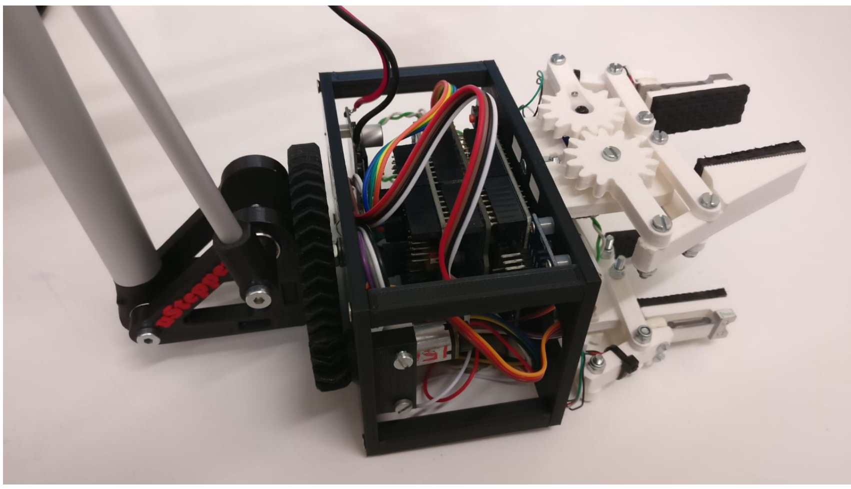


Figure 5.10: New end effector with dual gripper.

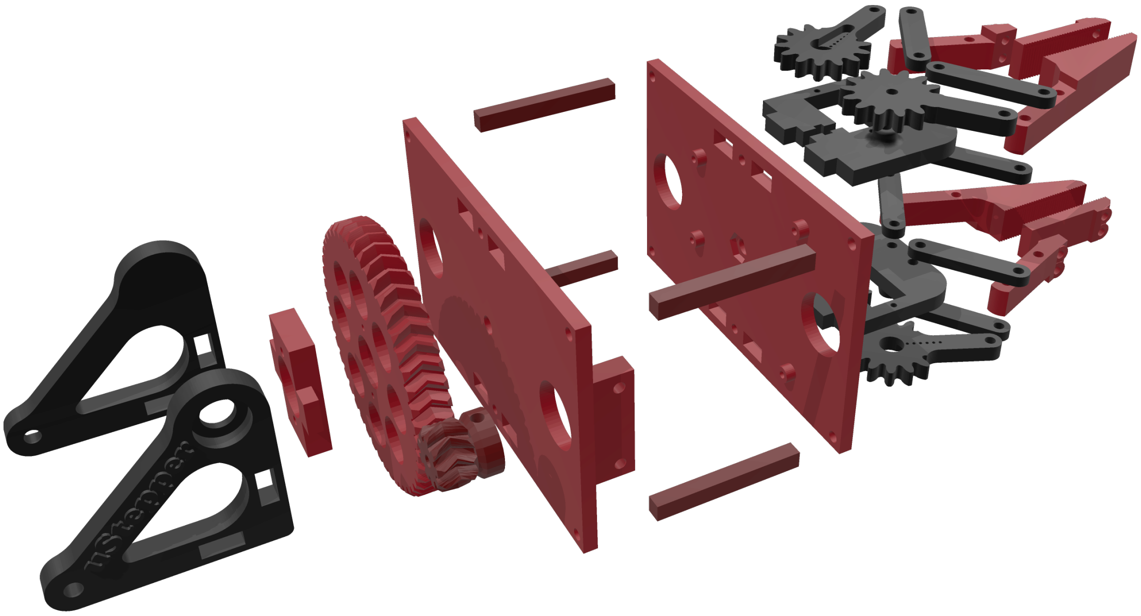


Figure 5.11: De-constructed 3D model of the end effector's pieces.

The YFRobot Sensor Shield, shown in the figure 5.12, is used to make connections easier the connections between the grippers' sensors and the Arduino Uno.

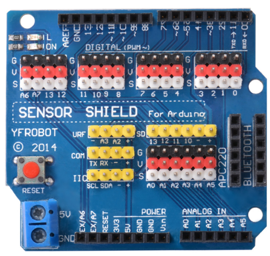


Figure 5.12: YFRobot Sensor Shield. Source [32].

The Arduino Uno, responsible for the control, is connected to the same laboratory's computer as the uStepper by USB, using the same communication platform and communication library "channels".

There is a servo motor attached to two of the three sets of gears and a DC motor attached to the third set. The servo motors are responsible for opening and closing the grippers and the DC motor makes the end effector rotate. The motors are connected to an Adafruit Motor Shield V2, shown in the figure 5.13, which allows the Arduino to control the motors. Each motor needs 6V to operate which comes from the same Adafruit Motor Shield that has a 19V to 6V converter, coming from one of the uStepper boards.

Inside the servo motors, there is a potentiometer. The potentiometer, usually connected to a control board inside the servo motor casing, indicates the angular position of the shaft's motor. To get this angle, the servo's control board was taken out and the potentiometer was directly connected to the Adafruit Motor Shield. This way is possible to get and give an angle directly to the motor.

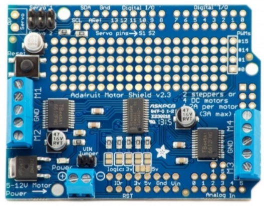


Figure 5.13: Adafruit Motor Shield V2. Source [33].

As the set of gears used to rotate the end effector have the same problem as the ones in the body manipulator, explained previously in the section 5.1 of this chapter, an accelerometer was placed to get the angle. Using an accelerometer eliminates the need to use a limit switch. As the accelerometer measures the difference between any linear acceleration in the reference frame and the gravity vector, applying the equation $angle = atan2(aceY, aceX)$, where aceY is the accelerometer component for the Y axis and aceX is the component for the X axis, it's possible to get the angle of the wrist rotation. The accelerometer is connected to the Arduino using I2C communication.

Force sensors are used to know the amount of force the grippers are applying in the objects. Using the sensors, makes it possible to close the gripper until a certain force is applied. This prevents the motors to keep trying to tighten the grippers when an object is already secure and prevent any possible damages to the servo motors. The grippers points are covered with a sponge like material to prevent the objects to slip from the gripper. These sensors are connected to an HX711 analog-to-digital converter (ADC), shown in the figure 5.14, which links to the Arduino by Serial Peripheral Interface (SPI). The HX711 amplifies the analog signal from the output voltage given by the force sensors and converts it into a digital signal which is sent to the Arduino.

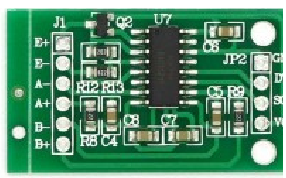


Figure 5.14: HX711: analog-to-digital converter. Source [34]

In the figure 5.15, it's possible to see how the components explained previously are interconnected resulting in the end effector's architecture.

Similarly to the functioning to move the body of the manipulator, the microcontroller Arduino receives a "Channels" message (further elaborated in the chapter 6) from the communication platform with a letter, corresponding to the type of action, and a value.

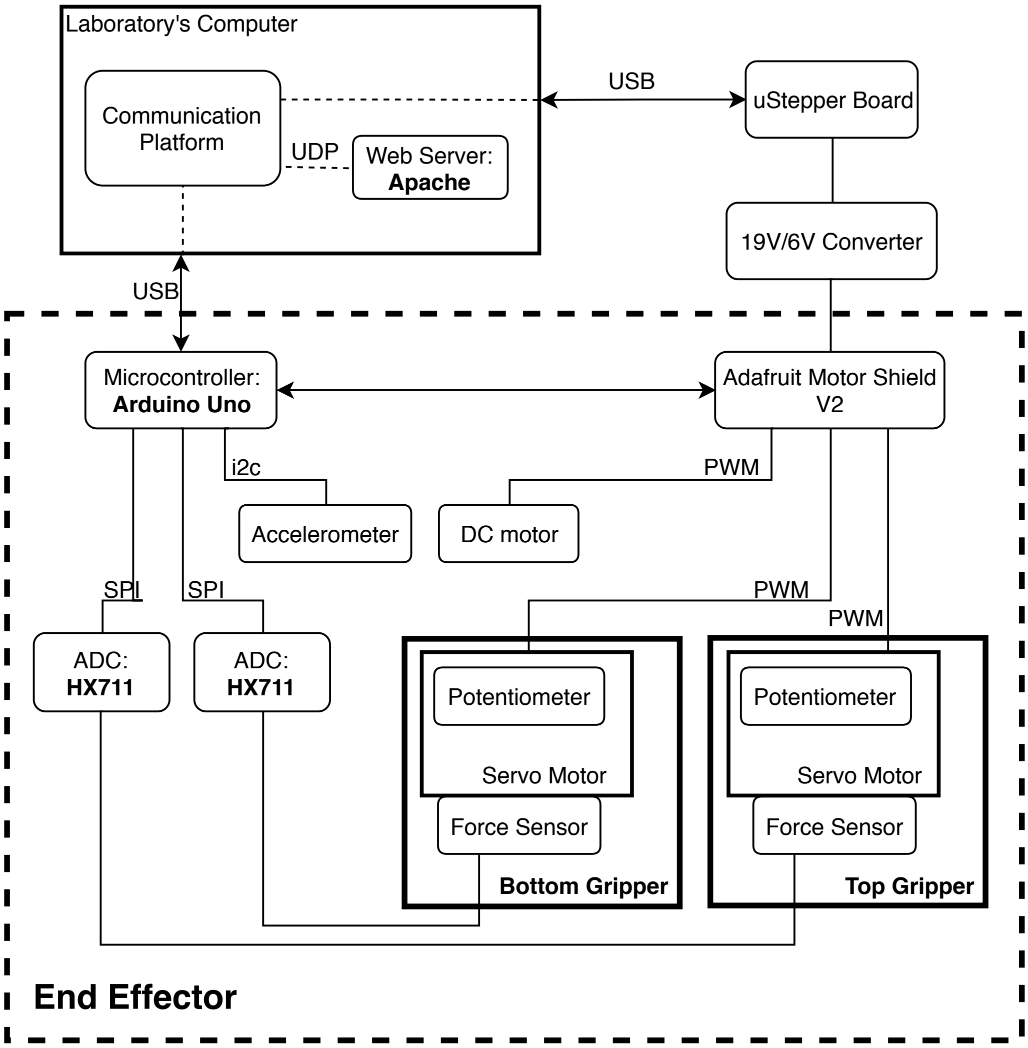


Figure 5.15: Architecture for the new end effector.

For the wrist rotation of the end effector, a reference angle or, in other words, the next position is received in the Arduino. Then, a pulse width modulated (PWM) signal is sent to the DC motor in order for it to start moving. A position controller, represented in the figure 5.16, is implemented in order to control the PMW signal given to the motor according to the angle error. This way, the end effector will rotate slower as it gets closer to the intended angle.

For the gripper to operate there are three available modes: angle mode, PWM mode and force mode. The algorithms for each of this modes are presented in the figure 5.17. The force mode is used when the action is to close the gripper because it prevents the servo motors to overwork when the gripper is holding an object. The maximum force applied can be set up to 200 gram force. The PWM mode is used to open the gripper as the force mode can't be used. To prevent strain on the motors, the gripper stays in a safe angle range. The PWM signal can also be set up to 255 (equivalent to a 100% duty-cycle). Even though the angle mode is implemented, it's not currently used in the system.

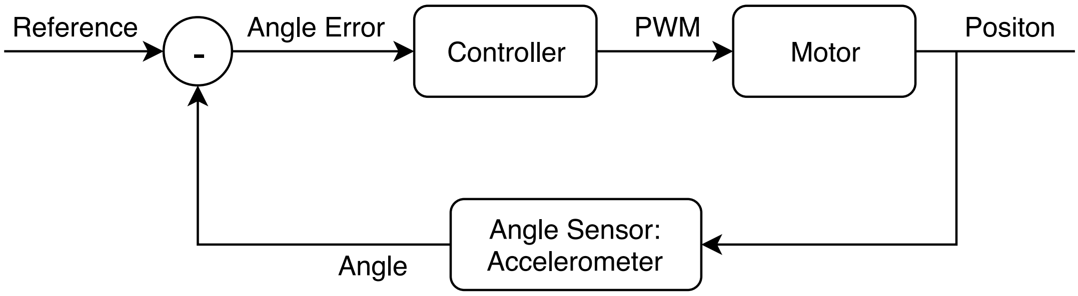


Figure 5.16: Position controller for the wrist rotation.

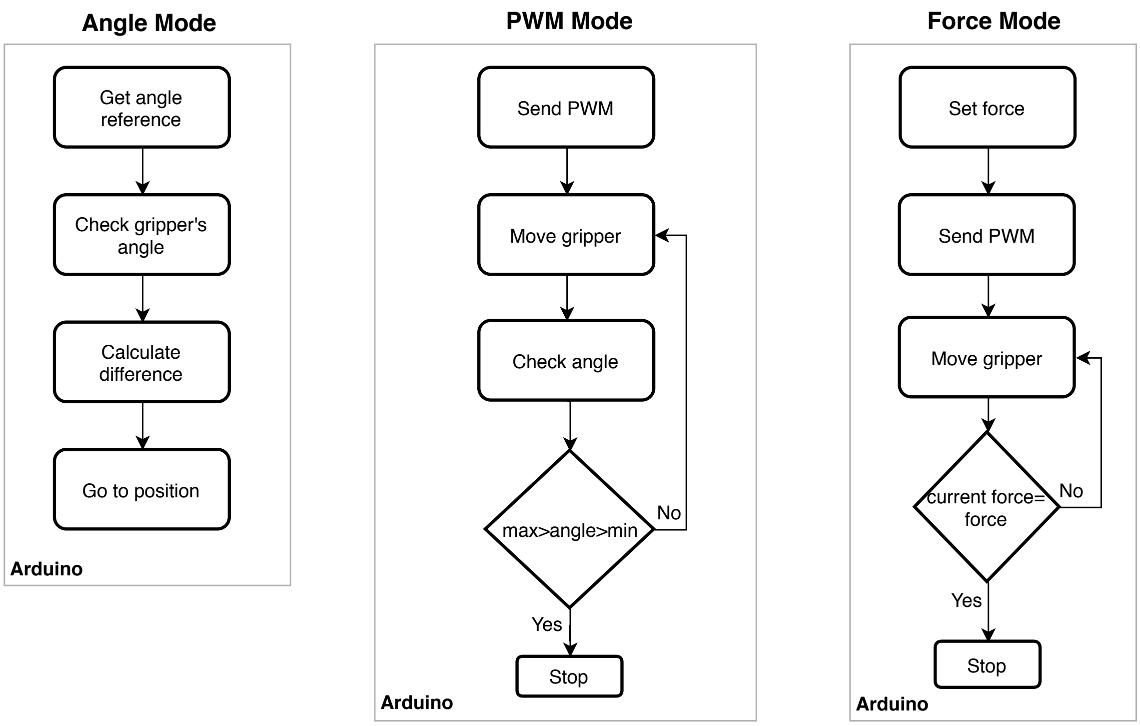


Figure 5.17: Algorithms for the three different working modes of the grippers.

5.3 Connecting the End Effector to the Body

Once both parts of the robotic manipulator where ready to be attached, a major setback was found. The motors aren't strong enough to move the arm in a smooth motion when the robotic manipulator has the end effector attached.

To try to solve this, springs are placed in the arm to support part of the weight of the end effector a shown in the figures . Despite helping the motors, the springs don't completely fix the weight problem.

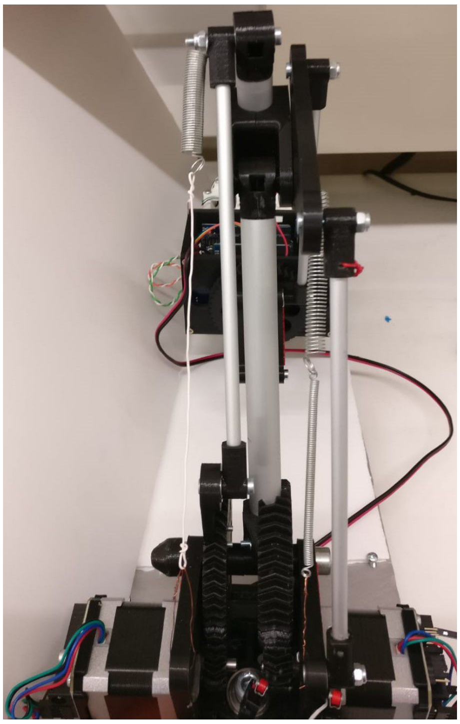


Figure 5.18: View of the springs on the robotic manipulator from the back.

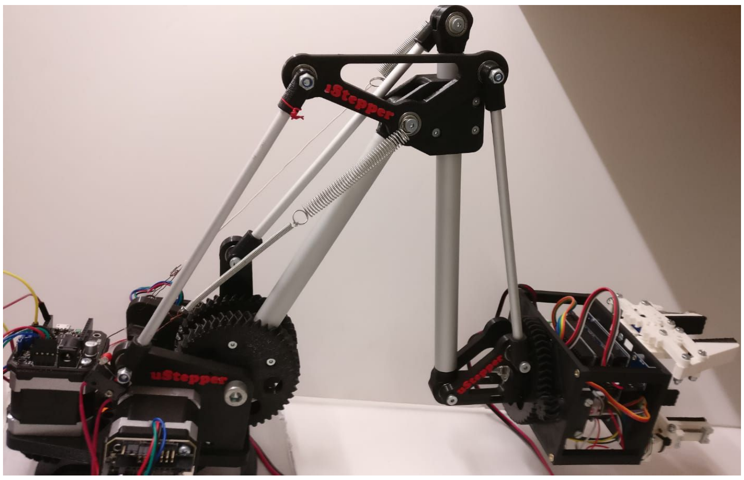


Figure 5.19: View of the springs on the robotic manipulator from the side.

A gearbox for each of the stepper motors is another possible solution. This would increase the strength of the motors while decreasing the speed. At the moment of writing this document, three gearboxes are being design and built.

A third option of increasing the current of the motors has been contemplated. However it has been set as the last option because the drivers of the uStepper boards can be easily damaged and higher current would increase the possibility of that happening.

Chapter 6

Communication Platform

A communication platform was created to send messages from the website to the robotic manipulator and vice-versa. An advantage of having this platform lies on the graphical user interface which is beneficial for the development and debugging process. Also, it's possible to have the web server hosted in a separated location from the laboratory if it's ever required.

The platform was designed using Lazarus: a cross-platform IDE for Free Pascal. Free Pascal is a compiler that runs on Windows, Mac OS X, and Linux. Using Lazarus allows also for the platform to run in any of the mentioned operating systems without the need to alter any code.

An existing serial port communication library called "Channels", developed by Professor Paulo Costa and available on his GitHub repository, was used as the protocol of communication between the platform and the microcontrollers (uStepper boards and Arduino). This library sends a message with a letter and a 4 bytes hexadecimal number. The letter symbolizes the channel, this is, the action to be performed. The number is the value of the action, for instance, the angle of a motor or the velocity. Also using this library, the microcontrollers can send a message back to the communication platform with the current position, for instance. This library was already implemented in Lazarus and prepared for Arduino like microcontrollers.

The communication platforms are two different programs, one for the motors of the body of the manipulator and another for the end effector.

The program corresponding to the body of the manipulator is called in three different instances (for each of the three uStepper boards) using different initializing files that changes certain variables. This different variables are the serial ports of each uStepper boards and UDP ports to which the messages are sent to and received from. An example of this interfaces can be seen on the figure 6.2 where it shows that the sets of gear corresponding to the motor and uStepper board 0 are not moving and the angle is 20°.

For the end effector, the communication interface can be seen in the figure 6.3. At the top of the interface, the force, angle and PWM signal for top servo motor, bottom servo motor and the wrist motor as well as the values given by the accelerometer can be observed. In the lower part of the interface there is a grid where the values to be sent to the Arduino are place and the command buttons.

It can be said that there are two different types of actions: movements of the body of the robotic manipulator and movements of the end effector. In the figure 6.1, a representation of the communication sequence (with examples) between the web server and the microcontrollers using the communication platforms is shown.

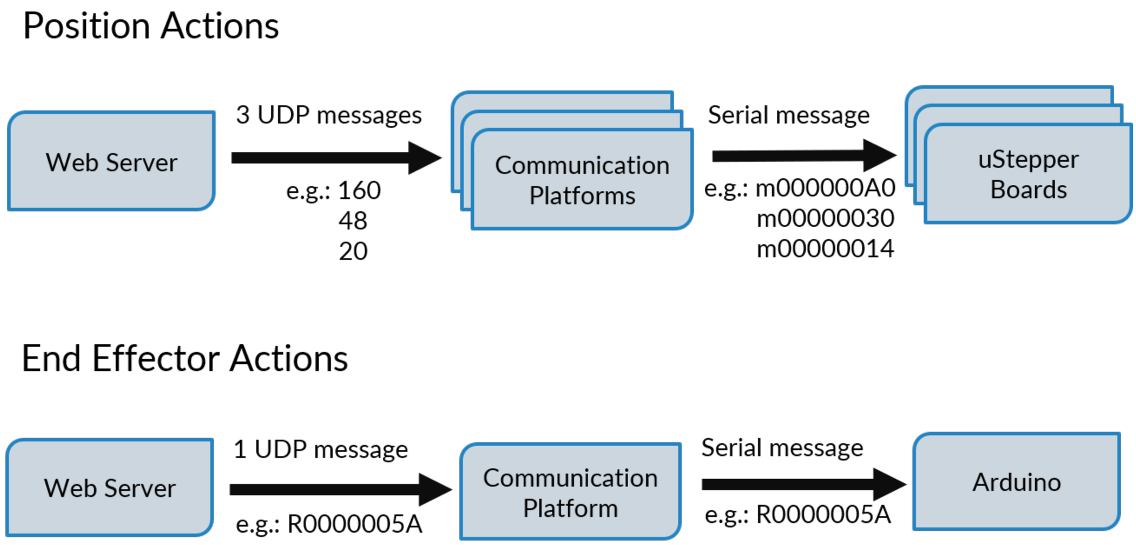


Figure 6.1: Representation of the communication sequence.

For a movement of the body of the robotic manipulator, the web server sends three UDP messages, each with a stepper motor angle, to each of the communication platforms. The platform transform each of the messages to a "Channels" message format. In this case, the letter "m" and the value of angle in a hexadecimal number of 8 bits. This angle is then sent to the uStepper boards by serial port. Finally, the uStepper boards will actuate in the stepper motors in order for the robotic manipulator to go to the wanted position.

For a movement of the end effector, the web server sends a single UDP message to the corresponding communication platform. This message is already in the "Channels" format (one letter and a hexadecimal number of 8 bits). The platform receives the message and sends it to the Arduino by serial port. Finally, the Arduino actuates in one of the three motors of the end effector in order to perform the action. As the end effector can perform various actions (rotate, opening and closing each of the grippers) different letters of the message corresponds to different actions. The hexadecimal number corresponds to an angle, a force or a PWM signal. An example of an end effector's message is shown in the figure 6.1, where "R0000005A" will rotate the end effector to 90°.

In the figure 6.4, the corresponding ports to which the messages are sent to for each of the microcontrollers is represented.

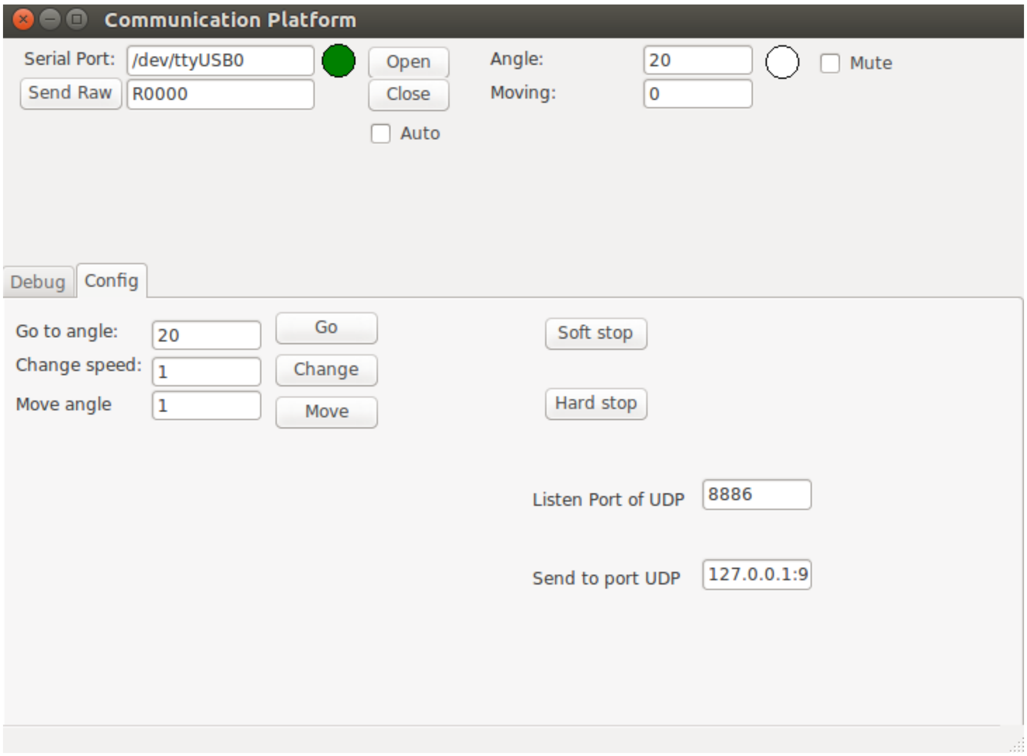


Figure 6.2: Interface of the communication platform for the uStepper 0.

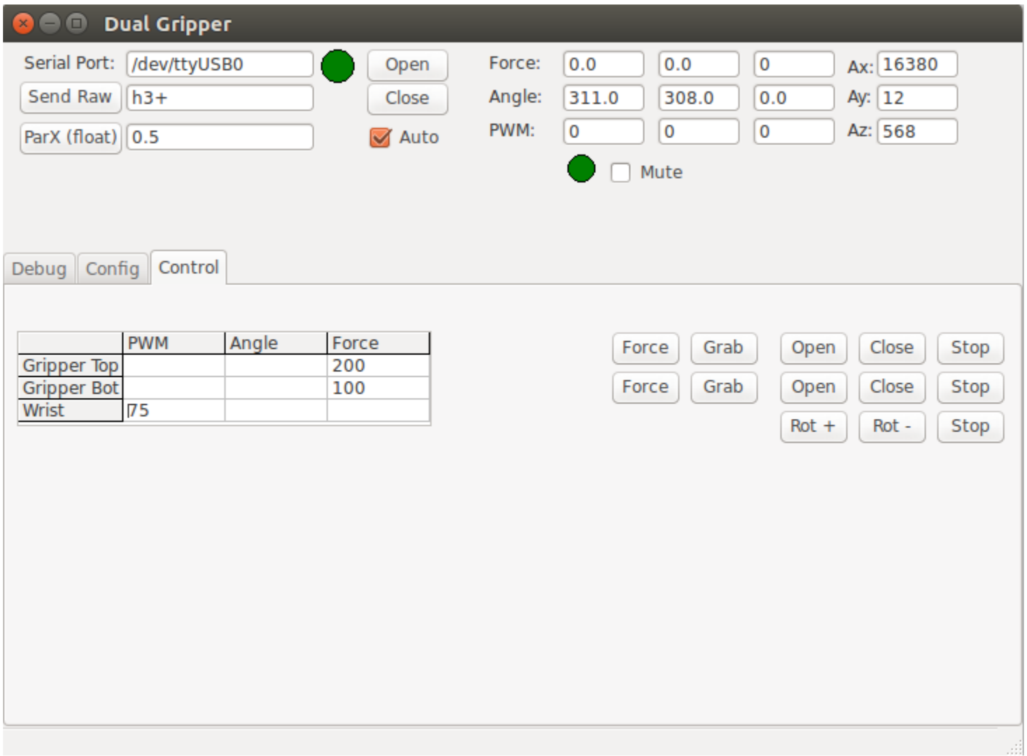


Figure 6.3: Interface of the communication platform for the end effector.

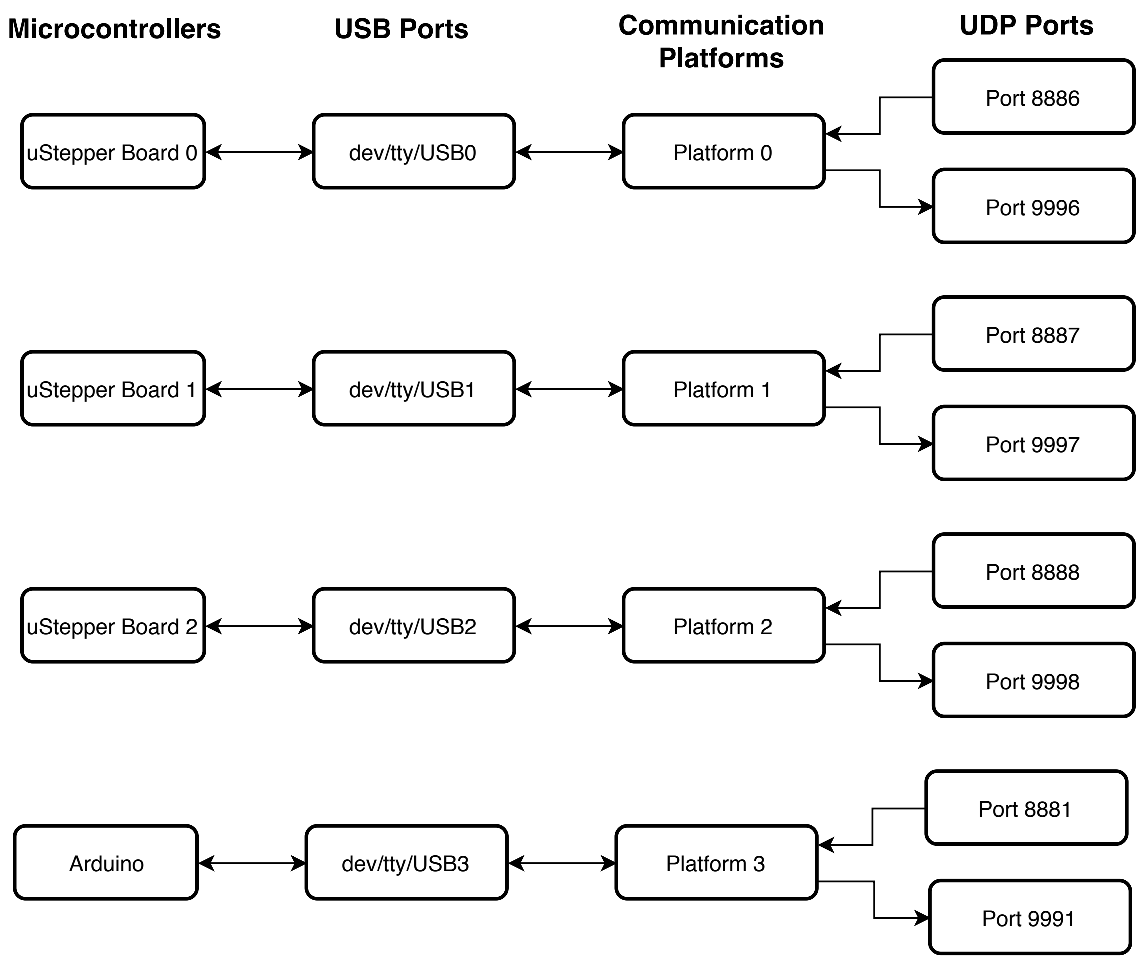


Figure 6.4: Representation of the connections between each element.

Chapter 7

Conclusions and Future Work

7.1 Compliance with the requirements

In the section 3.1, the requirements for the FEUP ChemLab were introduced.

Concerning the robotic manipulator's requirements, established in the subsection 3.1.1, not all of the requirements have been fulfilled. The requirements A.1, A.2 and A.3 are met, however the requirements A.4, A.5 and A.6 are not quite complete. The reason for this can be found in the fact that the robotic arm can't perform a smooth tilt motion due to the weight of the end effector. The robotic manipulator (without the end effector) can perform commands sent by the user and move to pre-defined positions. The end effector can also perform commands given by the user, pipette and transfer liquids. Be that has it may be, when the end effector is attached to the body, the robotic manipulator struggles to move and the A.4, A.5 and A.6 requirements can't be met.

Regarding the web interface's requirements, presented in the subsection 3.1.2, all of them were tested with success. To test the functionalities several computers (on the same local network) were used at the same time to access the website and perform the different possible actions. This being said, it can be considered that the interface has been successfully implemented and that it satisfies the future users' needs.

One of the goals of this project was for the FEUP ChemLab to be low cost so that schools are able to buy the system and implement it in their laboratories. In the table 7.1 the approximated cost and weight of the hardware components of the system is shown. The total cost for the hardware, at the time of writing this dissertation, was around 515 euros without taxes and without the cost of the 3D printed parts, the screws and the wires. As the FEUP ChemLab is an innovating system, it's hard to find any other similar systems to compare prices with.

Table 7.1: Approximated cost and weight of each component.

Component	Cost	Weight
uStepper Robotic Arm Kit	364 euros	1475 grams
Three limit switches	10 euros	5 grams
Arduino Uno	20 euros	28 grams
DC motor	10 euros	16 grams
Two servo motors	20 euros	36 grams
Adafruit Motor shield	20 euros	20 grams
YFrobot Sensor shield	6 euros	20 grams
Accelerometer	5 euros	20 grams
19/6V converter	5 euros	—
Two ADC HX711	5 euros	—
Two force sensors	10 euros	—
Hue HD Camera	40 euros	440 grams

7.2 Conclusions

As mentioned in the section 2.1.1.1, to validate the system the experiment in the appendix A would be performed. However, due to the problem explained previously, this experiment has not yet been performed using the robotic manipulator.

During the creation of this system, multiple problems have been found and solved. So it's believed that when this problem is solved that the FEUP ChemLab will be validated and will be considered a complete working system.

7.3 Contributions

With this project, an innovating system was created to help high school students and teachers. The FEUP ChemLab, constituted by a web interface, a web camera and a robotic manipulator with an end effector with wrist rotation and a force controlled dual gripper, creates the opportunity to perform more chemistry experiments during the school year. This is due to the fact that the system is a remote laboratory, meaning that the users don't need to be in the same room as the chemistry equipment and can complete an experiment after class time. Additionally, the system allows the users to monitor the experiments being performed by another user by viewing a live video feed of the workstation.

It should be noted that the manipulator used in the FEUP ChemLab is now able to transfer liquid from one recipient to another and pipette liquids. This are very used actions in chemistry experiments that previously (in the original uStepper Robotic Arm) were not possible.

The web interface is simple for any user to use without any extensive training or programming skills. The commands are intuitive, easy to define and to understand. The teachers can as well, easily add, manage and remove users.

In order to prevent user's mistakes, safety measures were implemented. The actions to be performed with the manipulator are always compared with the current position to make sure the movements are logical and physically safe to perform. For safety reasons, the speed and acceleration of the robotic manipulator is not controlled by the users. Furthermore, a teacher can "kick" the student using the robotic manipulator at the moment if deemed appropriated.

As far as studied, there is no other system like this one in the Portuguese schools, making the FEUP ChemLab a very important innovation that solves real problems.

7.4 Future Work

One of the most fundamentals tasks is to improve the tilt motion into a smoother and stronger movement, fulfilling the requirements A.4, A.5 and A.6. For this two things are planned:

- Increase the strength of the stepper motors using a gearbox.
- Reduce the end effector weight.

Besides that, some additional improvements can be made to the FEUP ChemLab.

- Reduce the video delay in the website.
- Add an automatic light source in the workstation.
- Add a public chat board on the website.
- Allow teachers to configure the new positions in a configuration page without the need to change the code.

Finally it would be important to test the FEUP ChemLab with a group of students and teachers who are unfamiliarized with the system and get their unbiased feedback.

Appendix A

Experiment's Protocol

Síntese do oxalato de ferro (II) diidratado

Realização experimental a microescala, utilizando o ácido ascórbico como redutor, proporções quase estequiométricas

A reacção entre o sulfato de ferro(II) heptaidratado e o ácido oxálico permite obter o oxalato de ferro(II) diidratado. A reacção pode representar-se pela equações químicas (1) e (2). Usa-se o ácido ascórbico para garantir que todo o ferro presente se encontra na forma de ferro (II).

$$Fe^{2+}\,(aq) + C_2O_4^{2-}\,(aq) + 2H_2O\,(l) \leftrightarrow FeC_2O_4.2H_2O\,(s) \qquad (1)$$

$$FeSO_4.7H_2O\,(aq) + H_2C_2O_4.2H_2O\,(aq) \leftrightarrow Fe(C_2O_4).2H_2O\,(s) + H_2SO_4\,(aq) + 7H_2O\,(l) \qquad (2)$$

1. Preparar uma solução de ácido oxálico contendo 124 mg (0,98 mmol) de ácido cristalino diidratado em 1,25 mL de água, num copo de 5 mL.

Figura 1 - Ácido oxálico

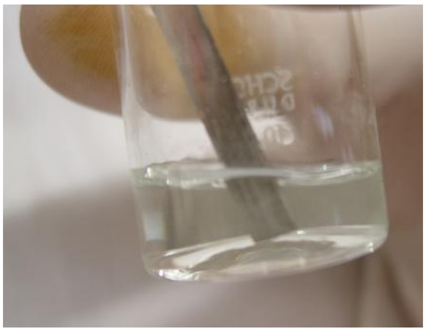


Figura 2 – Solução aquosa de ácido oxálico

2. Num copo de 10 mL dissolver 250 mg (0,90 mmol) de sulfato de ferro(II) heptaidratado em 1,25 mL de água desionizada e juntar 50 mg (0,28 mmol) de ácido ascórbico. Esperar 4 minutos (se a solução estiver a uma temperatura de 25ºC) para que se dê a redução de ferro(III) a ferro(II).

Figura 3 – Sulfato de ferro heptaidratado

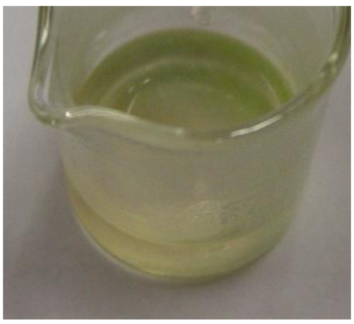


Figura 4 – Solução

3. Adicionar a solução de ácido oxálico à solução de sulfato de ferro. Forma-se um precipitado amarelo de oxalato ferroso.

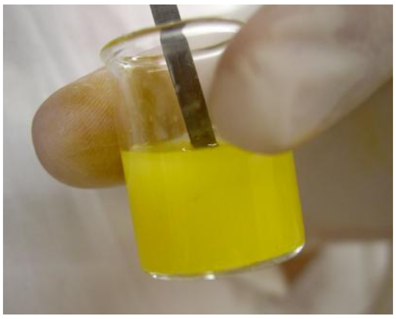


Figura 5 – Formação do precipitado de oxalato de ferro(II) diidratado

4. Agitar a mistura durante cerca de 10 minutos e deixar repousar. Retitrar o líquido sobrenadante com uma pipeta de Pasteur. Juntar cerca de 3 mL de água desionizada, mexer e retirar novamente o líquido sobrenadante com uma pipeta de Pasteur.

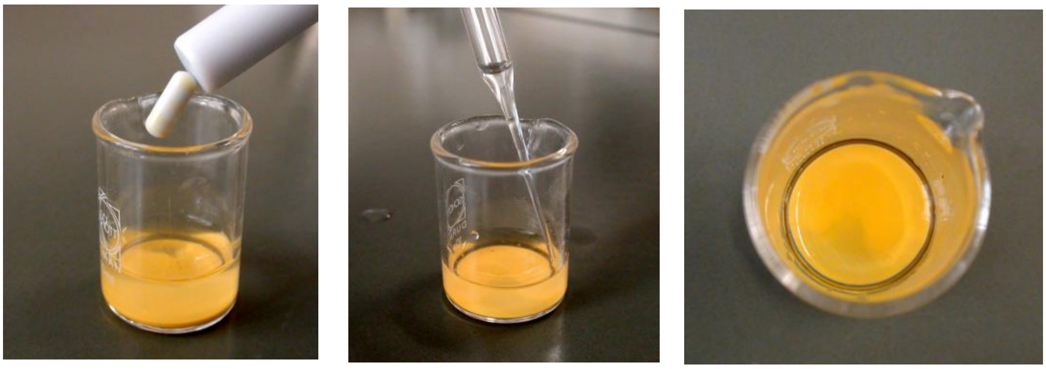


Figura 6 – Precipitado de oxalato de ferro(II) diidratado

5. Filtrar por sucção com um funil de Buchner e lavar o sólido com água desionizada.

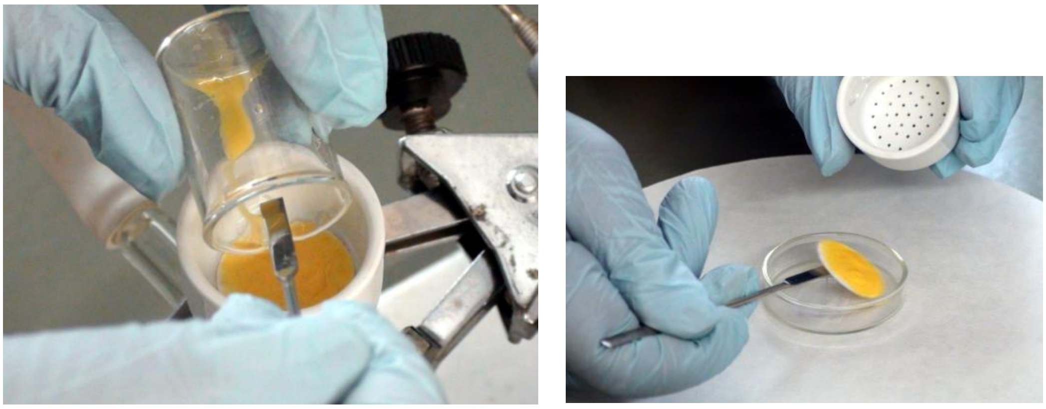


Figura 7 –Filtração

6. Colocar o produto no papel de filtro, num excicador, para secar. Depois de seco retirar o produto do papel de filtro e pesar.

Reagentes estequiométricos

- Sulfato de ferro (II) heptaidratado (CAS No. 7782-63.0)
MM ($FeSO_4.7H_2O$) = 278,01 g mol^{-1}

- Ácido oxálico diidratado (CAS No. 6153-56-6)
MM ($H_2C_2O_4.2H_2O$) = 126,07 g mol^{-1}

Substâncias auxiliares:

- Solventes
Água

- Outras substâncias auxiliares
- Ácido ascórbico (CAS No. 50-51-7)
MM ($H_8C_6O_6$) = 176,09 g mol^{-1}

Produto
Oxalato de ferro (II) diidratado
MM ($Fe(C_2O_4).2H_2O$) = 179,89 g mol^{-1}

Resíduos (condições próximas da estequiometria)
Ácido sulfúrico diluído
Ácido ascórbico
Ácido dehidroascórbico

Material
Copo 5 mL
Copo de 10 mL
Pipeta automática
Vidro de relógio
Caixa de Petri
Funil de Buchner
Equipamento para filtração por vácuo
Papel de filtro de filtração lenta
Pinça
Vareta de vidro
Espátula

Equipamento
Balança analítica ou semi-analítica
Placa de aquecimento com agitação magnética

References

[1] Equipa de Recursos e Tecnologias Educativas - Ministério da Educação. Iniciativa "Programação e Robótica no Ensino Básico", 2017, Visited on 11-01-2018. URL: http://erte.dge.mec.pt/noticias/iniciativa-programacao-e-robotica-no-ensino-basico.

[2] Ahmad Khanlari. Effects of educational robots on learning STEM and on students' attitude toward STEM. In *2013 IEEE 5th International Conference on Engineering Education: Aligning Engineering Education with Industrial Needs for Nation Development, ICEED 2013*, 2014.

[3] Maria Adelaide Ferreira d'Almeida Capela Ramos. Química Verde – Potencialidades e Dificuldades da sua Introdução no Ensino Básico e Secundário. Technical report, Universidade de Lisboa, 2009.

[4] Dennis L Hjeresen, David L Schutt, and Janet M Boese. Chemical Education Today Green Chemistry and Education Why Green Chemistry? *Journal of Chemical Education*, 77(12), 2000.

[5] Jing Ma and Jeffrey V. Nickerson. Hands-on, simulated, and remote laboratories. *ACM Computing Surveys*, 2006.

[6] Sinteses do oxalato de ferro(II) diidratado a microescala, 2008 , Visited on 11-01-2018. URL: http://educa.fc.up.pt/pedagogiadaquimicaverde/qv/sinteses.php?id=24.

[7] Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M. Petrović, and Kosta Jovanović. Virtual laboratories for education in science, technology, and engineering: A review. *Computers and Education*, 2016.

[8] James E. Corter, Sven K. Esche, Constantin Chassapis, Jing Ma, and Jeffrey V. Nickerson. Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories. *Computers and Education*, 2011.

[9] Luís Gomes and Seta Bogosyan. Current Trends in Remote Laboratories. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 56(12), 2009.

[10] Martyn Cooper and José M.M. Ferreira. Remote laboratories extending access to science and engineering curricular. *IEEE Transactions on Learning Technologies*, 2009.

[11] Ken Goldberg, Steve M. Gentner, Carl Sutter, and Jeff Wiegley. Mercury Project: A feasibility study for Internet robots. *IEEE Robotics and Automation Magazine*, 2000.

[12] Diego López, Raquel Cedazo, Francisco Manuel Sánchez, and José María Sebastián. Ciclope robot: Web-based system to remote program an embedded real-time system. *IEEE Transactions on Industrial Electronics*, 2009.

[13] Olawale Akinwale, Lawrence Kehinde, Kayode P Ayodele, A M Jubril, Olutola P Jonah, and Xuemin Chen. AC 2009 - 1179 : A LABVIEW - BASED ON - LINE ROBOTIC ARM FOR STUDENTS ' LABORATORY. *American Society for Engineering Education,*, 2009.

[14] Roland Szabó and Aurel Gontean. Remotely commanding the lynxmotion AL5 type robotic arms. *21st Telecommunications forum TELFOR*, 2013.

[15] Martin Jaanus, Andres Udal, Vello Kukk, and Kadri Umbleja. Implementation of the Robot Arm in the Interactive Learning Environment. 2014.

[16] Herminio Martínez-García. Implementation of a remote laboratory for distance training in robotic applications. In *Proceeding - 2015 IEEE International Conference on Industrial Informatics, INDIN 2015*, 2015.

[17] Chaitanya Nutakki, Asha Vijayan, Hemalatha Sasidharakurup, Bipin Nair, Krishnashree Achuthan, and Shyam Diwakar. Low-cost robotic articulator as an online education tool: Design, deployment and usage. In *International Conference on Robotics and Automation for Humanitarian Applications, RAHA 2016 - Conference Proceedings*, 2017.

[18] Lynxmotion. AL5D PLTW Robotic Arm Kit, Visited on 28-01-2018. URL: http://www.lynxmotion.com/p-1019-al5d-pltw-robotic-arm-kit.aspx.

[19] Lynxmotion AL5D PLTW Robotic Arm Kit, Visited on 28-01-2018. URL: https://www.robotshop.com/en/lynxmotion-al5d-pltw-robotic-arm-kit.html.

[20] Lynxmotion. SSC-32U USB Servo Controller, Visited on 28-01-2018. URL: http://www.lynxmotion.com/p-1032-ssc-32u-usb-servo-controller.aspx.

[21] uStepper, Visited on 29-01-2018. URL: http://www.ustepper.com/index/.

[22] uStepper GitHub, Visited on 29-01-2018. URL: https://github.com/uStepper/uStepper.

[23] UStepper. uStepper Rev . B product description - DATA SHEET.

[24] Intel NUC, Visited on 22-01-2018. URL: https://www.intel.com/content/www/us/en/products/boards-kits/nuc.html.

[25] Chris Stobing. What is a NUC PC, and Should You Get One?, 2016, Visited on 22-01-2018. URL: https://www.howtogeek.com/231978/htg-explains-what-is-a-nuc-pc-and-should-you-get-one/.

[26] Jason Baker. What is a Raspberry Pi?, 2015, Visited on 23-01-2018. URL: https://opensource.com/resources/raspberry-pi.

[27] Raspberry Pi, Visited on 23-01-2018. URL: https://www.raspberrypi.org/.

[28] Eric Huang. To USB or Not to USB, 2012, Visited on 28-01-2018. URL: https://blogs.synopsys.com/tousbornottousb/2012/12/06/an-explanation-of-the-usb-video-class-uvc/.

[29] Hue. Hue HD Camera, Visited on 01-06-2018. URL: https://huehd.com/products/hue-hd-camera/.

[30] PHP Manual. URL: http://php.net/manual/en/.

[31] ROBOT ARM, Visited on 15-06-2018. URL: http://ooznest.co.uk/Robot-Arm.

[32] YFRobot Sensor Shield, Visited on 15-06-2018. URL: https://www.trab.dk/da/arduino-shields/230-yfrobot-sensor-shield.html.

[33] Motor Shield Adafruit v2, Visited on 15-06-2018. URL: https://www.filipeflop.com/produto/motor-shield-adafruit-v2-arduino/.

[34] Modulo Sensor Duplo-canal HX711, Visited on 15-06-2018. URL: http://www.botnroll.com/pt/sensores/1039-modulo-sensor-duplo-canal-hx711.html.